NON-INTRUSIVE EXTENSION OF A GENERALIZED FINITE ELEMENT METHOD FOR MULTISCALE PROBLEMS TO THE ABAQUS ANALYSIS PLATFORM

BY

JULIA A. PLEWS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Carlos Armando Duarte

Abstract

Several classes of important engineering problems – in this case, problems exhibiting sharp thermal gradients – have solution features spanning multiple spatial scales of interest and, therefore, necessitate advanced *hp* finite element discretizations. Although *hp*-FEM is unavailable off-the-shelf in many predominant commercial analysis software packages, a novel method is proposed herein which is used to introduce these capabilities via the generalized FEM with global-local enrichments (GFEM^{gl}) [14] non-intrusively in Abaqus, a popular, general-purpose FEA platform. Numerical results show that the techniques utilized allow for accurate resolution of localized thermal features on structural-scale meshes without *hp*-adaptivity or the ability to account for very localized loads in the FEM framework itself. This methodology enables the user to take advantage of all the benefits of both *hp*-FEM discretizations and the appealing features of many available CAE/FEA software packages in order to obtain optimal convergence for challenging multiscale problems.

Acknowledgments

Most of all, I would like to express my gratitude to my advisor, Professor C. Armando Duarte, for providing me with the opportunity to conduct research in the field of finite element methods. His dedication to high-quality work and consistently practical, forward-thinking perspective during the process were instrumental in the completion of this thesis.

I would like to acknowledge my colleague and officemate Dr. Patrick J. O'Hara, whose work has provided a solid foundation for my own research, and who offered many helpful resources and suggestions throughout my studies; also, the late Dr. Jerônymo P.A. Pereira, whose contributions to the development of methodologies examined herein were also crucial to this project's success.

Additionally, I would like to thank Drs. Thomas Eason and Ravinder Chona of the United States Air Force Research Laboratory for their helpful contributions and encouragement over the past two years of my work on this project.

I must thank the Department of Civil & Environmental Engineering at the University of Illinois at Urbana-Champaign for fellowship support of my studies, and the Midwest Structural Sciences Center in the Department of Computational Science and Engineering at the University of Illinois, supported by the U.S. Air Force Research Laboratory, Air Vehicles Directorate, under contract number FA8650-06-2-3620, for providing me with research funding during the pursuit of my Master's degree.

Of course, I am always grateful to my colleagues, friends, and especially my family, who have also been tremendously supportive of all my efforts.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation & Background

A growing number of problems encountered in engineering practice today require consideration of phenomena encompassing multiple spatial scales of interest. One example of particular interest – and which is part of the motivation behind this work – lies in the structural analysis of hypersonic flight vehicles. At very high airspeeds, rapid variations in the density and temperature of the compressible flow lead to shock impingements on the skin of the vehicle. Interactions between shock waves, typically occurring most severely on the leading edge of the aircraft wing, have been shown to lead to very intense, localized thermomechanical loads. Characterization of these complicated effects itself has been an active research topic [7, 21, 22, 30, 41–43]. A comprehensive, historical overview of the challenges posed by aero-thermal-mechanical effects in hypersonic structures and their importance is given in [40]. In the scope of this study, however, only the thermal component of loadings is considered.

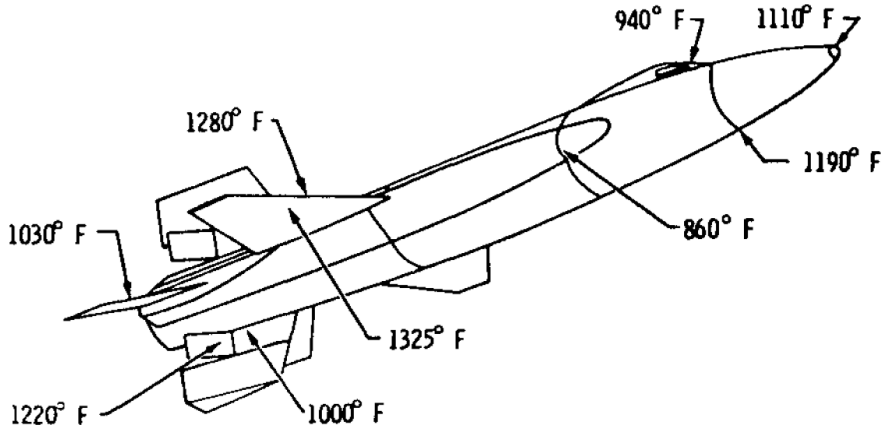


Figure 1.1: Measured temperatures on the skin of a North American X-15 spaceplane from an actual hypersonic flight at Mach 5.0 (Cf. [40]).

Based on experimental investigations, intense, localized heat fluxes can be concentrated on an area just microns in width – that is, many orders of magnitude smaller than the structure. However, these loadings may have a drastic effect on the overall behavior of the structure as well as the neighborhood of the loading itself, and thus the two disparate

scales of interest may not in general be considered separately.

In current design practice, commercial finite element analysis (FEA) software packages are commonly used to predict the response of, for example, the hypersonic structures of interest. Optimal finite element discretizations for this class of problems require state-of-the-art localized, adaptive mesh refinement combined with high-order polynomial approximations [34, 39]. Unfortunately, performing such *hp*-adaptivity in available FEA software is often prohibitively difficult or, in many cases, even impossible. Additionally, treatment of sharp, localized thermal loads requires special attention and cannot in general be computed automatically in conventional FEA software. More importantly, if these loadings are applied on meshes designed to capture only the global response of the structure, the error of the finite element solution may be large even far away from the localized features due to so-called pollution error [2, 33].

1.2 Objectives

To alleviate this issue, it is proposed to introduce an *hp* Generalized Finite Element Method (GFEM) to an existing FEA platform without any code modifications, which is demonstrated here using the commercial software Abaqus. Abaqus was chosen due to its popularity in the engineering community, its heat transfer capabilities, and its robust scripting interface for data input, program execution, and output of results, which was utilized heavily in this project. The method used, however, is quite general and can be applied to a variety of FEA software.

This non-intrusive approach offers many benefits, circumventing the need for *hp*-adaptivity in Abaqus and enabling accurate computation of sharp thermal loads on structural-scale meshes. As a result, it will be shown to yield optimal convergence for this class of problems. Additionally, the methodology will be able to provide a great deal of flexibility in handling a variety of multiscale analysis cases for the same structural model of interest. While this work focuses on intense, localized heat sources, the methodology is relevant to a much broader range of problems exhibiting multiscale phenomena without modifications to the overall approach.

1.3 Outline

In chapters to follow, the overall methodology and sample results from the implementation are enumerated. The problem of interest is defined, and the proposed solution methodology, a generalized finite element method, is explained in some detail in Chapter 2. The approach taken to implement this method non-intrusively in Abaqus is additionally described in Chapter 3, focusing on the solution algorithm as well as the computational features used in executing Abaqus analyses. Finally, results from three sample problems of interest are given in Chapter 4, which serve to demonstrate the primary advantages of using the non-intrusive implementation in practical applications – specifically,

convergence rates, error levels, and flexibility of the method.

All GFEM analyses are carried out with the help of the Illinois Scientific and Engineering Toolbox (ISET), a research code originally developed by Professor C. Armando Duarte and continuously maintained and enhanced by him and a team of his students at the University of Illinois at Urbana-Champaign. A `twoSolver` framework has been built into this code in order to make it cooperate nicely with a wide variety of external analysis codes – including itself – specifically for the purpose of the non-intrusive implementation studied in the scope of this project.

Chapter 2

Problem Definition

2.1 Definition of the Boundary Value Problem

2.1.1 Strong Form

Consider a domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega = \Gamma^u \cup \Gamma^f \cup \Gamma^c$, where $\Gamma^u \cap \Gamma^f = \emptyset$, $\Gamma^u \cap \Gamma^c = \emptyset$, and $\Gamma^c \cap \Gamma^f = \emptyset$. The strong form of the governing partial differential equation is given by Poisson's equation,

$$\nabla(\boldsymbol{\kappa}\nabla u) = -Q(\boldsymbol{x}) \quad \text{in} \quad \Omega, \tag{2.1}$$

herein given the physical interpretation of heat transfer, where $u(\boldsymbol{x}) \equiv u(x_1, x_2, x_3)$ is the temperature field, $\boldsymbol{\kappa}$ is the thermal conductivity tensor, and $Q(\boldsymbol{x}) \equiv Q(x_1, x_2, x_3)$ is the internal heat source. Boundary conditions prescribed on $\partial\Omega$ are given by

$$\begin{aligned} u &= \bar{u} \quad \text{on} \quad \Gamma^u \\ -\boldsymbol{\kappa}\nabla u \cdot \boldsymbol{n} &= \bar{f} \quad \text{on} \quad \Gamma^f \\ -\boldsymbol{\kappa}\nabla u \cdot \boldsymbol{n} &= \alpha(u - u_\infty) \quad \text{on} \quad \Gamma^c \end{aligned} \tag{2.2}$$

where $\boldsymbol{n}$ is the outward unit normal vector to Γ^f and Γ^c, and $\bar{f}$ and $\bar{u}$ are prescribed normal heat flux and temperature, respectively. α is the convection coefficient, and u_∞ is the free-stream temperature for convective conditions.

2.1.2 Weak Form

Utilizing the principle of virtual work, Poisson's equation can be satisfied in a "weak" or integral sense, that is, for all $v(\boldsymbol{x}) \in V$, find $u(\boldsymbol{x}) \in U$ such that

$$\int_\Omega \nabla u \boldsymbol{\kappa} \nabla v \, d\Omega - \int_{\Gamma^c} \alpha u v \, d\Gamma = \int_\Omega Q v \, d\Omega + \int_{\Gamma^f} \bar{f} v \, d\Gamma - \int_{\Gamma^c} \alpha u_\infty v \, d\Gamma, \tag{2.3}$$

The left hand side of (2.3) is commonly known as the bilinear form, $B(u,v)$. The energy norm associated with this bilinear form is then

$$||u||_E := \sqrt{B(u,u)}. \tag{2.4}$$

The spaces of functions U and V in (2.3) are defined as

$$\begin{aligned} U &= \{u(\boldsymbol{x}) : B(u,u) < \infty; \quad u = \bar{u} \quad \text{on} \quad \Gamma_u\}, \\ V &= \{v(\boldsymbol{x}) : B(v,v) < \infty; \quad v = 0 \quad \text{on} \quad \Gamma_u\}. \end{aligned}$$

2.2 Spatial Discretization of the Boundary Value Problem

2.2.1 Finite Element Approximation

The temperature field u may be approximated as

$$u(\boldsymbol{x}) \approx \boldsymbol{N}(\boldsymbol{x}) \cdot \boldsymbol{d} \quad \text{on} \quad \Omega, \tag{2.5}$$

a Ritz approximation, where $\boldsymbol{N}(\boldsymbol{x})$ has $\varphi_\alpha(\boldsymbol{x})$, finite element shape functions, or interpolating functions defined at each node in a mesh covering domain Ω. $\boldsymbol{d}$ are the corresponding nodal degrees of freedom, or coefficients multiplying shape functions $\boldsymbol{N}$.

Substituting the approximation of the temperature field (2.5) in the weak formulation (2.3) such that $u \approx \boldsymbol{N}\boldsymbol{d}$ and $v \approx \boldsymbol{N}\delta\boldsymbol{d}$, the Galerkin form of the problem is obtained:

$$\delta\boldsymbol{d}^T \int_\Omega \boldsymbol{B}^T \boldsymbol{\kappa} \boldsymbol{B}\boldsymbol{d} \, d\Omega - \delta\boldsymbol{d}^T \int_{\Gamma^c} \alpha \boldsymbol{N}^T \boldsymbol{N}\boldsymbol{d} \, d\Gamma = \delta\boldsymbol{d}^T \int_\Omega \boldsymbol{N}^T Q \, d\Omega + \delta\boldsymbol{d}^T \int_{\Gamma^f} \boldsymbol{N}^T \bar{f} \, d\Gamma - \delta\boldsymbol{d}^T \int_{\Gamma^c} \boldsymbol{N}^T \alpha u_\infty \, d\Gamma, \tag{2.6}$$

where

$$\boldsymbol{B}(\boldsymbol{x}) := \nabla \boldsymbol{N}(\boldsymbol{x}).$$

In order to satisfy this for all $v(\boldsymbol{x}) \in V$, the nontrivial solution corresponding to $\delta\boldsymbol{d} \neq \boldsymbol{0}$ must be found, resulting in the linear system of equations

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}, \tag{2.7}$$

where

$$\begin{aligned} \boldsymbol{K} &:= \int_{\Omega} \boldsymbol{B}^T \boldsymbol{\kappa} \boldsymbol{B}\, d\Omega - \int_{\Gamma^c} \alpha \boldsymbol{N}^T \boldsymbol{N}\, d\Gamma \\ \boldsymbol{f} &:= \int_{\Omega} \boldsymbol{N}^T Q\, d\Omega + \int_{\Gamma^f} \boldsymbol{N}^T \bar{f}\, d\Gamma - \int_{\Gamma^c} \boldsymbol{N}^T \alpha u_\infty\, d\Gamma. \end{aligned}$$

2.2.2 GFEM Approximations

The generalized finite element method (GFEM) [1, 3, 15, 32, 38] is a Galerkin method based on discretization spaces defined using the concept of a partition of unity (*PoU*). Partition of unity methods originated from the work of Babuška et al. [1, 3, 28] as well as Duarte and Oden [13, 17–19, 32]. The extended finite element method, or X-FEM, is another example of a method based upon *PoU* concepts with many similarities to the GFEM [4, 29]. The usefulness of the GFEM hinges on the idea that the partition of unity can be enriched, or combined with local function approximation spaces built around *a-priori* knowledge about the solution of a given problem.

In the GFEM, standard finite element shape functions φ_α are chosen as the partition of unity, since $\varphi_\alpha,\ \alpha = 1,\ldots,N$, in a mesh covering a domain Ω with N nodes are such that $\sum_{\alpha=1}^{N} \varphi_\alpha(\boldsymbol{x}) = 1$ for all $\boldsymbol{x}$ in Ω. A GFEM shape function $\phi_{\alpha i}$ is then computed as the product of the FEM *PoU* φ_α and an enrichment function $L_{\alpha i}$,

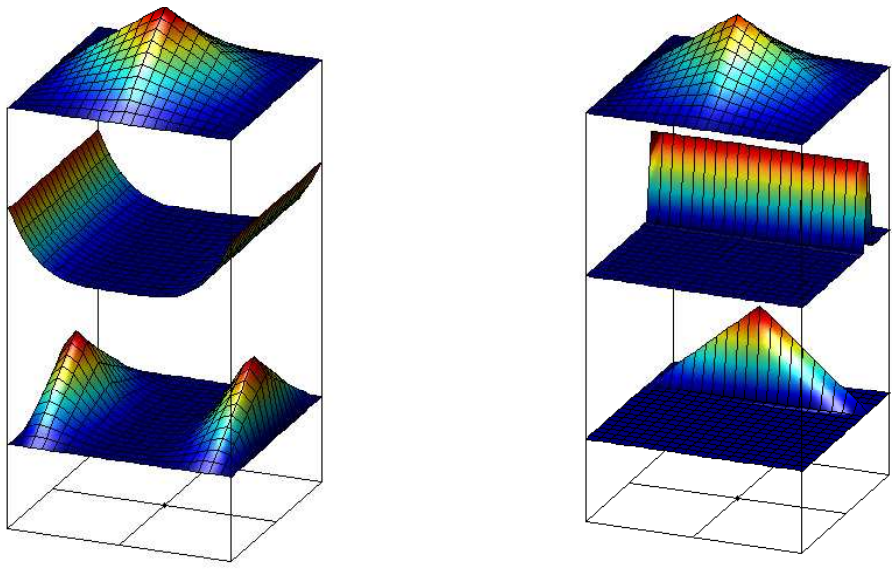
$$\phi_{\alpha i}(\boldsymbol{x}) = \varphi_\alpha(\boldsymbol{x}) L_{\alpha i}(\boldsymbol{x}) \qquad (\text{no summation on } \alpha), \tag{2.8}$$

where α is a node in the FE mesh. Figure 2.1 illustrates shape function construction for various types of enrichments.

2.2.3 GFEM$^{\mathbf{gl}}$

Although special GFEM enrichment functions may be designed to accommodate analytical solution characteristics, for instance, in the case of fracture mechanics [16] or modeling of polycrystalline structures [37], in many cases *a-priori* knowledge of the solution behavior is limited. Moreover, especially in large problems, performing *hp*-adaptivity on a structural-scale mesh in order to account for localized solution features may add many degrees of freedom to the problem and, thus, may prove much too computationally expensive.

The GFEM with global-local enrichment functions (GFEM$^{\text{gl}}$) [14, 33], however, allows for on-the-fly, numerical creation of custom enrichments via the solution of smaller, *hp*-adapted local problems which enclose features of interest in the structural-scale (global) domain. Thus, expensive mesh refinements and localized, high-order polynomial enrichments need not be done in the global domain itself, and only a few degrees of freedom are added to the global



(a) (b)

Figure 2.1: Construction of a GFEM shape function – from top to bottom, φ_α, the FEM partition of unity, $L_{\alpha i}$, an enrichment function, and $\phi_{\alpha i}$, the resulting shape function for (a) a polynomial enrichment function, and (b) a custom, non-polynomial enrichment.

problem as a result of the numerically-built enrichment functions. The most basic GFEM[gl] solution procedure is comprised of three main steps – a coarse-scale initial solution, extraction of local problems, and enrichment and reanalysis of the global problem based on local solutions.

Initial Global (*IG*) Problem An initial, coarse-scale analysis is first performed on the global problem on $\bar{\Omega}_G = \Omega_G \cup \partial\Omega_G$, yielding initial solution u^0. The initial global problem is formulated as, for all $v^0 \in X_G(\Omega_G)$, find $u^0 \in X_G(\Omega_G)$ such that

$$\int_{\Omega_G} \nabla u^0 \boldsymbol{\kappa} \nabla v^0 d\Omega + \eta \int_{\Gamma_G^u} u^0 v^0 d\Gamma - \int_{\Gamma_G^c} \alpha u^0 v^0 d\Gamma = \int_{\Omega_G} q v^0 d\Omega + \int_{\Gamma_G^f} \bar{f} v^0 d\Gamma + \eta \int_{\Gamma_G^u} \bar{u} v^0 d\Gamma - \int_{\Gamma_G^c} \alpha u_\infty v^0 d\Gamma, \tag{2.9}$$

where η is a predetermined penalty parameter for enforcement of Dirichlet boundary conditions, and $X_G(\Omega_G)$ is a GFEM discretization of $H^1(\Omega_G)$.

Local Problem(s) The initial global solution resulting from the linear system of equations implied by (2.9) is then used directly as a Dirichlet boundary condition in a local problem. The local domain is comprised of $\bar{\Omega}_L = \Omega_L \cup \partial\Omega_L$, a subdomain of Ω_G, which in practice is taken to be a user-specified, small subset of finite elements extracted (copied) from the global problem, upon which *hp*-adaptivity may then be performed. Selection of local domains and adaptive mesh refinement may also be automated by using *a-posteriori* error estimates on the initial global solution, u^0. The

local problem is formulated as follows: for all $v_L \in X_L(\Omega_L)$, find $u_L \in X_L(\Omega_L)$ such that

$$\int_{\Omega_L} \nabla u_L \boldsymbol{\kappa} \nabla v_L d\Omega + \eta \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \Gamma_G^f)} u_L v_L d\Gamma - \int_{\partial\Omega_L \cap \Gamma_G^c} \alpha u_L v_L d\Gamma$$
$$= \eta \int_{\partial\Omega_L \setminus (\partial\Omega_L \cap \partial\Omega_G)} u^0 v_L d\Gamma + \eta \int_{\partial\Omega_L \cap \Gamma_G^u} \bar{u} v_L d\Gamma + \int_{\Omega_L} q v_L d\Omega + \int_{\partial\Omega_L \cap \Gamma_G^f} \bar{f} v_L d\Gamma - \int_{\partial\Omega_L \cap \Gamma_G^c} \alpha u_\infty v_L d\Gamma, \quad (2.10)$$

where $X_L(\Omega_L)$ is again a GFEM discretization of $H^1(\Omega_L)$. The initial global and local problem steps comprise a procedure akin to the global-local FEM [12, 20, 31].

Enriched Global (*EG*) Problem The solution of the local problem may not in general provide an accurate estimate of the true solution to the global problem, for instance, as was demonstrated for the case of fracture mechanics of multiple interacting cracks [27], or as will be shown by an example heat transfer problem in Section 4.3, due to inaccurate boundary conditions on the local problem.

However, the solution of the local problem is quite useful as an enrichment function for the GFEM near the localized feature(s) of interest. Thus, taking the global-local analysis one step further, the solution u_L yielded by the local problem is used in the GFEM$^{\text{gl}}$ to build so-called global-local shape functions

$$\phi_\alpha^{gl}(\boldsymbol{x}) = \varphi_\alpha(\boldsymbol{x}) u_L(\boldsymbol{x}) \quad (2.11)$$

numerically, which are in turn added to the global approximation space and utilized to enrich and re-solve the global problem as defined in (2.9). This step is termed the enriched global (*EG*) problem, the solution of which is denoted hereafter u^E.

Improvement of Local Problem Solutions In order to obtain a better approximation of global solution behavior from a local problem, resulting in overall better enriched global solutions, a simple strategy has been identified. The flexibility of the GFEM$^{\text{gl}}$ allows for the selection of a local domain which is arbitrarily larger than the so-called "enrichment zone," or the area of the mesh which is actually enriched with the local problem solution. This approach is known as selecting a "buffer zone," and it serves to damp out the effect of poor boundary conditions on the local problem [26].

Chapter 3

Non-Intrusive Implementation of the GFEM^{gl} in a FE Solver

In recent years, non-intrusive methodologies such as the one proposed here have been actively investigated in order to enable a smooth transition of advanced GFEM/X-FEM discretizations to application in popular and venerable commercial FEA codes such as Abaqus. Most predominantly, a number of software packages and implementations which provide GFEM/X-FEM fracture mechanics modeling capabilities in Abaqus (or other comparable FEA platforms) have been presented. Similar to the motivation behind this implementation, these packages serve to extend meshing flexibility to legacy codes while offering comparable – or better! – solution approximations.

For instance, Giner et al. [25], Shi et al. [35, 36], and Xu and Yuan [44, 45] propose implementing the X-FEM by making use of `UEL`, user element subroutines, or `UMAT`, user materials, within Abaqus. However, in the view of the author, use of `UEL` routines in Abaqus suffers from some major drawbacks. For instance, user element subroutines may suffer from a lack of customizable or powerful features because they are not necessarily independent finite element codes in and of themselves. Additionally, specifying user elements requires knowledge of, for example, global-local enrichment zone information *a-priori*; that is, in a `UEL` approach, the user must manually select elements in the global model which will possess X-FEM enrichments ahead of the analysis, meaning that the user elements are an integral part of the Abaqus input, and that the global model must then be altered for each separate analysis case or localized feature of interest.

Gendre et al. [23, 24] propose a non-intrusive, nonlinear FEM implementation in Abaqus which is somewhat similar to the GFEM methodology introduced here, where a patch of elements containing a localized plastic region is in fact "exactly" extracted from the global problem by a Schur complement method. However, their approach is limited by the fact that the boundary of the local patch of extracted elements must exactly match the global problem mesh where the local patch is inserted; that is, no refinement can be performed along the boundary of the local patch. Due to the use of a partition of unity, this is not an issue in the GFEM^{gl}, which is demonstrated on a sample mesh in Figure 3.1. Also, the Schur complement of the *local* degrees of freedom – as opposed to the global degrees of freedom, the approach adopted here – is computed, which in general comprises many degrees of freedom and thus proves very expensive for large global problems.

The procedure which will be described here takes a different approach from the aforementioned philosophies.

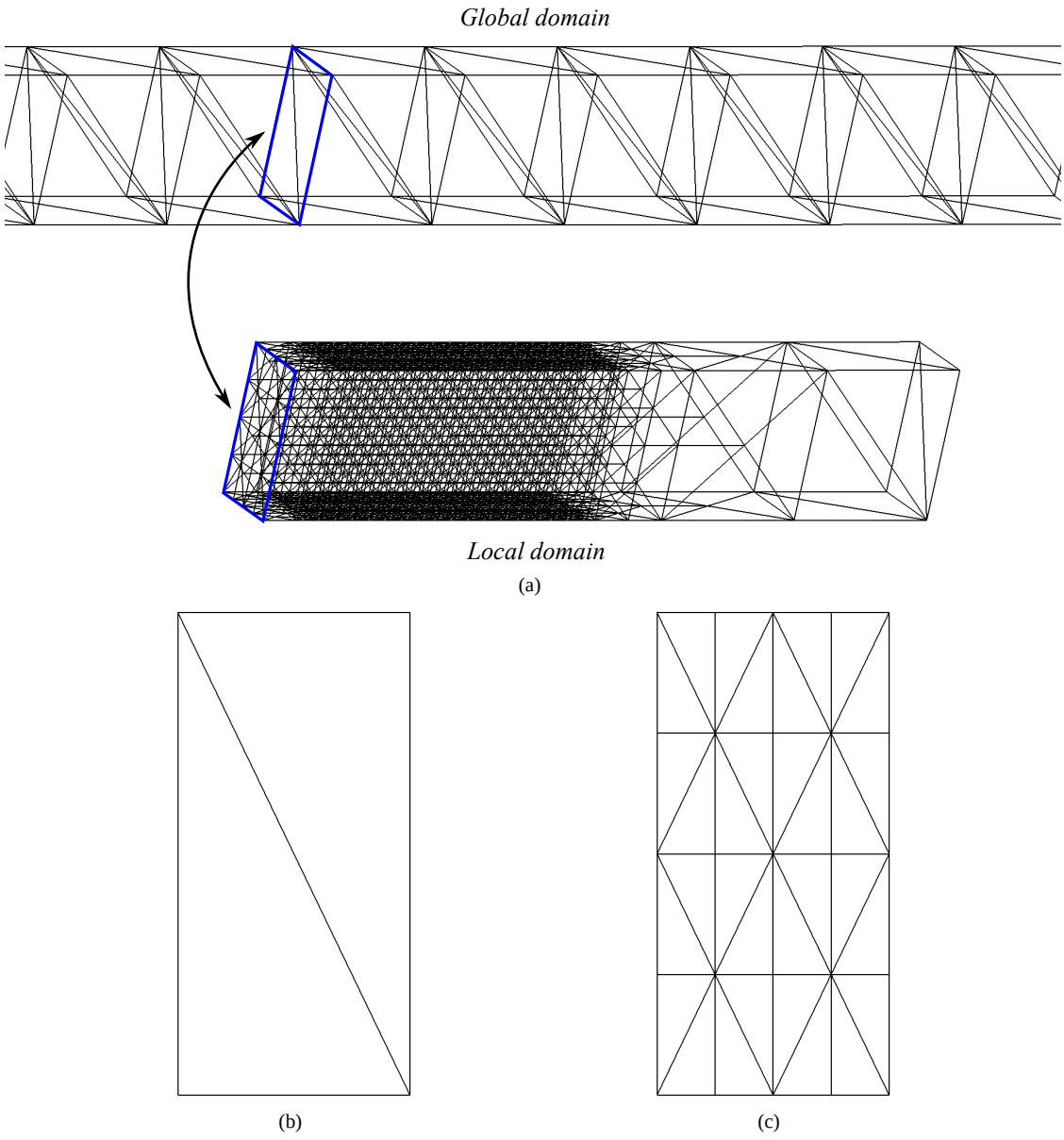


Figure 3.1: Demonstration of the possibility for arbitrarily refined meshes at the local domain boundary, irrespective of the matching global problem mesh. The use of the local solution as an enrichment function, sewn together with the global problem approximation by a partition of unity, allows for much more flexibility than, for example, "exact" extraction and solution of a localized subset of the domain. Figure 3.1(a) shows the corresponding mesh faces in the global and the local domain, respectively, while (b) and (c) show a zoom-in on the difference in mesh refinement between the two.

That is, in this case, the non-intrusive implementation consists of (i) Abaqus as a standalone code, and (ii) another standalone, in-house GFEM code. The finite element system of equations is partitioned, and portions are solved in each code separately (explained further in Section 3.1). Thus, the only communication which takes place between the two is controlled by (iii) a converter code designed specifically for this purpose (detailed in Section 3.4). The approach implemented herein is perhaps most similar to Bordas and Moran [5] (who are using the commercial code EDS-PLM/I-DEAS®) and is in a sense the "inverse" procedure to the Schur complement method for scale-bridging described in Gendre et al. [23, 24], which will be explained much further in the following section.

3.1 Partitioned System of Equations

The GFEM^{gl} by its very nature is readily extensible to a multiple-solver implementation; one standard FE solver (referred to here as **FE-S**) is made to handle the coarse-scale global problem, and another GFEM solver (**GFE-S**) orchestrates the analysis by handling the local and enriched global problem aspects. The GFEM solution of the enriched global problem, u^E, can be partitioned as

$$u^E = \tilde{u}^0 + u^{gl} = \left[\begin{array}{cc} \boldsymbol{N}^0 & \boldsymbol{N}^{gl} \end{array} \right] \left[\begin{array}{c} \underline{\tilde{\boldsymbol{u}}}^0 \\ \underline{\boldsymbol{u}}^{gl} \end{array} \right], \tag{3.1}$$

where $\boldsymbol{N}^0$ has standard FEM shape functions and $\boldsymbol{N}^{gl}$ has the global-local shape functions defined in (2.11). Vectors $\underline{\tilde{\boldsymbol{u}}}^0$ and $\underline{\boldsymbol{u}}^{gl}$ have global and global-local enrichment degrees of freedom, respectively. Then, the gradient of the temperature field is defined as

$$\nabla u^E = \left[\begin{array}{cc} \boldsymbol{B}^0 & \boldsymbol{B}^{gl} \end{array} \right] \left[\begin{array}{c} \underline{\tilde{\boldsymbol{u}}}^0 \\ \underline{\boldsymbol{u}}^{gl} \end{array} \right]. \tag{3.2}$$

The resulting system of equations in the enriched global problem, as formulated in and implied by (2.9), $\boldsymbol{K}\underline{\boldsymbol{u}}^E = \boldsymbol{f}$ may then also be partitioned as

$$\left[\begin{array}{cc} \boldsymbol{K}^0 & \boldsymbol{K}^{0,gl} \\ \boldsymbol{K}^{gl,0} & \boldsymbol{K}^{gl} \end{array} \right] \left[\begin{array}{c} \underline{\tilde{\boldsymbol{u}}}^0 \\ \underline{\boldsymbol{u}}^{gl} \end{array} \right] = \left[\begin{array}{c} \boldsymbol{f}^0 \\ \boldsymbol{f}^{gl} \end{array} \right], \tag{3.3}$$

where

$$\boldsymbol{K}^0 := \int_{\Omega_G} \left(\boldsymbol{B}^0\right)^T \boldsymbol{\kappa} \boldsymbol{B}^0 \, d\Omega$$

is computed by **FE-S**, and

$$\begin{aligned} \boldsymbol{K}^{0,gl} &:= \int_{\Omega_L} \left(\boldsymbol{B}^0\right)^T \boldsymbol{\kappa} \boldsymbol{B}^{gl} \, d\Omega \\ \boldsymbol{K}^{gl} &:= \int_{\Omega_L} \left(\boldsymbol{B}^{gl}\right)^T \boldsymbol{\kappa} \boldsymbol{B}^{gl} \, d\Omega. \end{aligned}$$

are computed in **GFE-S**. The solution to System (3.3) can then be found by static condensation on $\underline{\boldsymbol{u}}^{gl}$, since, in general, $dim\left(\underline{\tilde{\boldsymbol{u}}}^0\right) \gg dim\left(\underline{\boldsymbol{u}}^{gl}\right)$. From the first equation in System (3.3),

$$\begin{aligned} \underline{\tilde{\boldsymbol{u}}}^0 &= \left(\boldsymbol{K}^0\right)^{-1} \boldsymbol{f}^0 - \left(\boldsymbol{K}^0\right)^{-1} \boldsymbol{K}^{0,gl} \underline{\boldsymbol{u}}^{gl} \\ &= \underline{\boldsymbol{u}}^0 - \boldsymbol{S}^{0,gl} \underline{\boldsymbol{u}}^{gl}, \end{aligned} \tag{3.4}$$

where $\boldsymbol{K}^{0,gl}$ are known as "pseudo-loads,"

$$\boldsymbol{S}^{0,gl} := \left(\boldsymbol{K}^0\right)^{-1} \boldsymbol{K}^{0,gl}$$

are known as "pseudo-solutions" corresponding to the pseudo-loads, and

$$\underline{\boldsymbol{u}}^0 := \left(\boldsymbol{K}^0\right)^{-1} \boldsymbol{f}^0$$

is the initial, coarse-scale solution. Both $\boldsymbol{S}^{0,gl}$ and $\underline{\boldsymbol{u}}^0$ may be computed by **FE-S** using forward and backward substitution on a factorization of the coarse-scale global stiffness matrix, $\boldsymbol{K}^0$. Because $\boldsymbol{K}^0$ does not change between initial global and enriched global problems, the factorization of $\boldsymbol{K}^0$ may also, if possible, be stored in **FE-S** after the initial global problem step and reused in the enriched global problem in order to reduce computational cost as well as solution time. From Equation (3.4) and the second equation in System (3.3),

$$\begin{aligned} \boldsymbol{K}^{gl} \underline{\boldsymbol{u}}^{gl} &= \boldsymbol{f}^{gl} - \boldsymbol{K}^{gl,0} \underline{\tilde{\boldsymbol{u}}}^0 \\ &= \boldsymbol{f}^{gl} - \boldsymbol{K}^{gl,0} \left(\underline{\boldsymbol{u}}^0 - \boldsymbol{S}^{0,gl} \underline{\boldsymbol{u}}^{gl}\right). \end{aligned} \tag{3.5}$$

Through some rearranging, this leads to

$$\underbrace{\left(\boldsymbol{K}^{gl} - \boldsymbol{K}^{gl,0} \boldsymbol{S}^{0,gl}\right)}_{\hat{\boldsymbol{K}}_{gl}} \underline{\boldsymbol{u}}^{gl} = \underbrace{\boldsymbol{f}^{gl} - \boldsymbol{K}^{gl,0} \underline{\boldsymbol{u}}^0}_{\hat{\boldsymbol{f}}_{gl}}, \tag{3.6}$$

so the solution corresponding to global-local degrees of freedom comes directly from the solution of

$$\hat{\boldsymbol{K}}^{gl}\underline{\boldsymbol{u}}^{gl} = \hat{\boldsymbol{f}}^{gl},$$

where $\hat{\boldsymbol{K}}^{gl}$ can be interpreted as the Schur complement of $\boldsymbol{K}^0$.

From all the above, it is evident that the static condensation algorithm requires only the exchange of pseudo-loads and pseudo-solutions between **FE-S** and **GFE-S**, making this approach extensible to almost any FEA software package. The algorithm described here is also illustrated graphically in Figure 3.2.

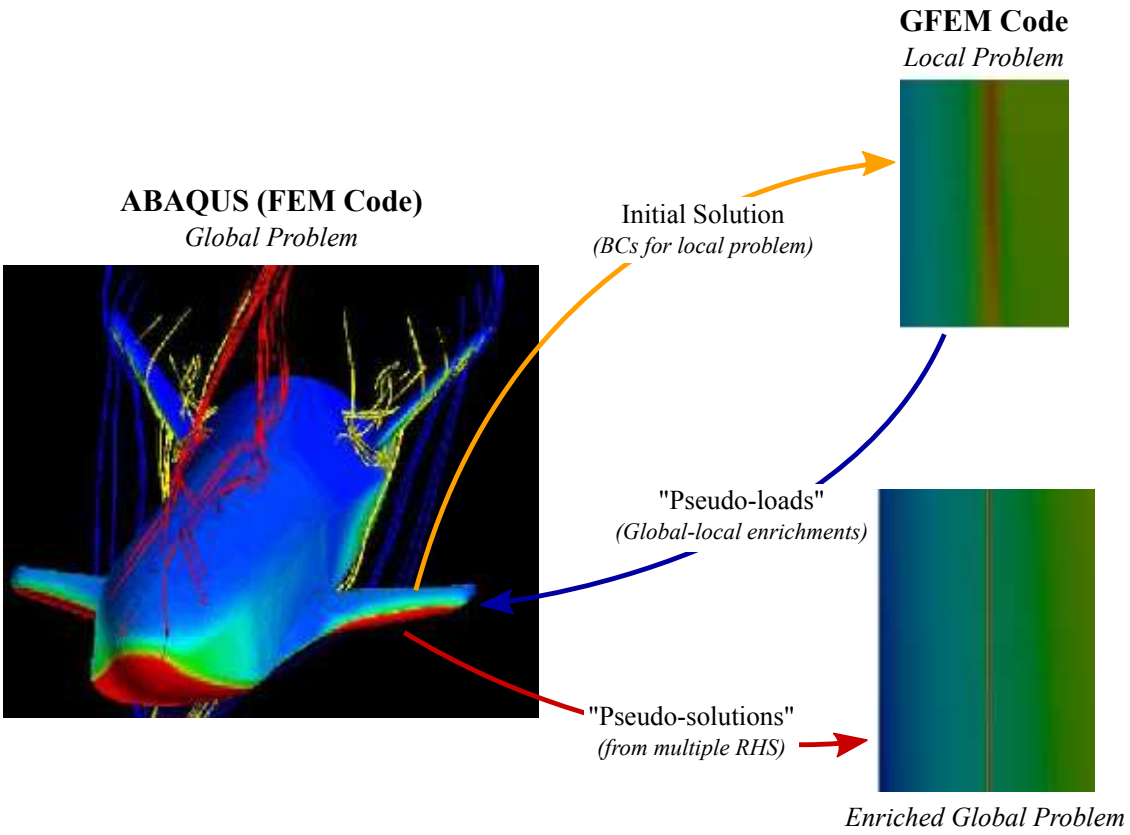


Figure 3.2: Illustration of non-intrusive implementation – exchange of "pseudo-loads" and "pseudo-solutions" between FEM and GFEM codes.

3.2 Treatment of Rough Loads

Standard FEM solvers cannot typically handle sharp heat fluxes applied on meshes designed to capture only the coarse-scale component of the solution. Furthermore, application of such sharp loadings on coarse meshes may cause error in the finite element solution to propagate even far from the localized feature (so-called "error pollution") [2, 33].

Thus, the global load vector, $\boldsymbol{f}^0$ above, will be decomposed as

$$\boldsymbol{f}^0 = \boldsymbol{f}_R^0 + \boldsymbol{f}_S^0,$$

where $\boldsymbol{f}_R^0$ is the "rough," sharp, localized portion of the load, and $\boldsymbol{f}_S^0$ is the remaining "smooth" portion. Although **FE-S** is able to compute $\boldsymbol{f}_S^0$ without difficulty, the additional sharp load $\boldsymbol{f}_R^0$ must be numerically integrated using **GFE-S**.

As was examined in [33], it will be shown that applying *only* the smooth loading, $\boldsymbol{f}_S^0$, in the initial global problem not only gives a satisfactory estimate of the solution to develop local problem boundary conditions but also eliminates aforesaid error pollution effects.

Then, it follows that an approach must be developed to accurately compute the total load on the global structure – that is, including the true, sharp load features. Taking advantage of information available from the local problem step, the highly refined, *hp*-adapted local domain elements may be used as so-called integration elements in the global problem, coupled with a high-order numerical quadrature rule, to recompute the total load, $\boldsymbol{f}^0$, in **GFE-S** just prior to the enriched global problem. The improved global load vector is then passed to **FE-S** alongside the pseudo-loads, and this right hand side is solved during the enriched global problem phase (further explanation of this procedure will be given in Section 3.4). The results of this "improved" global analysis effectively supersede the solution obtained from the coarse-scale, initial global step, $\underline{\boldsymbol{u}}^0$, and are used in place of the initial global solution in the static condensation algorithm for the enriched global solution. This procedure is critical in order to obtain optimal convergence in the enriched global solution. The effects of these strategies on numerical results will be shown for a sample problem in Section 4.2.

In typical FEA software, even finding a way to apply the proper sharp loading – analytically defined or otherwise – can be a nontrivial task. It will be demonstrated further in Section 4.2 that standard, commercial finite element codes may not necessarily have any built-in protocol for handling sharp, user-defined loadings on coarse meshes, providing additional benefit to the proposed approach.

3.3 Adoption of a TET10 Partition of Unity

Even when using global-local enrichments, on a coarse, global mesh, a linear approximation of the global solution to many classes of problems often may not yield satisfactory error levels. In the GFEM, it is common practice to choose a linear partition of unity and enrich these "hat" functions with higher-order polynomials in order to improve global approximations [15, 32]. Similarly, in the state-of-the-art *hp*-version of the finite element method, arbitrarily high-order shape functions can be hierarchically added to elements in a finite element mesh [39].

However, in many available commercial FEA codes, for three-dimensional analyses, the choices available to the user for the order of the approximation is often severely limited. In Abaqus, for example, only linear (TET4) or quadratic (TET10) tetrahedral elements are implemented for heat transfer simulations. Moreover, quadratic elements in Abaqus are implemented through standard Lagrangian finite element shape functions, adding additional nodes to the tetrahedral element; thus, the partition of unity used in the GFEM^{gl} enriched global problem must similarly accommodate the ten-noded tetrahedral element. In GFEM^{gl} local problems, however, because the local approximation spaces are immaterial to the global one, TET10 elements can be directly converted to TET4 elements upon which *hp*-adaptivity may be performed just as before.

Some examples of the improvement in enriched global solutions and convergence results thanks to the use of a TET10 global approximation in Abaqus models will be shown in Chapter 4.

3.4 Communication Between Abaqus and GFE-S

As mentioned in Section 3.1, each analysis using the non-intrusive implementation of the GFEM^{gl} in Abaqus (hereafter Abaqus+GFEM^{gl}) is orchestrated by **GFE-S**, which executes continuously throughout the process. All communication which occurs between Abaqus and the GFEM code – the exchange of the initial global solution, pseudo-loads, and pseudo-solutions – is facilitated by a converter code written in a combination of Python (the scripting language of Abaqus/CAE [10, 11]) and C++. It follows from this two-standalone-solver approach that identical global models, or job files, must be present for both Abaqus and **GFE-S**. Two model files must be written for Abaqus: one for the initial global analysis (henceforth `abaqus.inp`), and another for the enriched global problem, which will contain pseudo-loads (`abaqus_nRHS.inp`). Abaqus/CAE is utilized here for its `.odb` binary output database format, which contains all user-requested output in a very conveniently organized, hierarchical data structure and can be read and converted directly by an Abaqus Python script (referred to here as the Python converter code).

Initial Global Problem The procedure for the initial global problem step is as follows:

(i) Call Abaqus from **GFE-S** to execute model file `abaqus.inp` via Python script.

(ii) When problem is done executing, execute Python converter code to write initial global results $\underline{\boldsymbol{u}}^0$ from the `.odb` file to output file readable by **GFE-S**.

(iii) Read initial solution in **GFE-S**.

Local Problem(s) Once the initial global problem solution is read in **GFE-S**, execution continues, and local problems can subsequently be extracted and solved, as specified by the user. The procedure for a local problem is as

follows:

(i) Extract user-specified local domain Ω_L, and apply initial solution u^0 from Abaqus as Dirichlet boundary conditions.

(ii) Perform *hp*-adaptivity as requested by the user and solve local problem.

(iii) Using local solution u_L, compute global-local shape functions ϕ_α^{gl} as in Equation (2.11), and use these to assemble $\boldsymbol{K}^{gl}$ and $\boldsymbol{f}^{gl}$ corresponding to global-local dofs, and $\boldsymbol{K}^{0,gl}$, pseudo-loads to be passed to Abaqus, as defined in Section 3.1.

(iv) Recompute global load vector $\boldsymbol{f}^0$, using local mesh for integration.

(v) Write out pseudo-loads and recomputed global load vector (including the sharp load features) to a file.

Enriched Global Problem The enriched global problem requires multiple communications between **GFE-S** and Abaqus. In this step, because in general multiple pseudo-loads $\boldsymbol{K}^{0,gl}$ must be solved, Abaqus's built-in capability to solve multiple right hand sides as part of the same job is utilized, so that assembly and factorization of the global stiffness matrix $\boldsymbol{K}^0$ in Abaqus need not be computed over again for each pseudo-load. However, the commonly used Abaqus command `*LOAD CASE` built to handle multiple load vectors is geared primarily for elasticity/structural analysis problems and is currently unavailable for heat transfer problems. Thus, an alternative approach using multiple steady-state analysis steps was identified to enable Abaqus to compute solutions for multiple right hand sides in this case. This and other limitations will be discussed further in Section 5.2. The enriched global procedure is done as follows:

(i) Read pseudo-loads and global load vector from **GFE-S** in converter code, and write each right hand side therein to a separate analysis step in an input file stub readable by Abaqus.

(ii) Call Abaqus and execute enriched global model `abaqus_nRHS.inp` (written prior to the analysis), including file containing multiple right hand sides (*not* known prior to the analysis) written above.

(iii) When problem is done executing, execute Python converter code to write resulting pseudo-solutions $\boldsymbol{S}^{0,gl}$ and recomputed global solution $\underline{\boldsymbol{u}}^0$ from the `.odb` file to output file readable by **GFE-S**.

(iv) Read output file in **GFE-S** and compute the global degrees of freedom, $\underline{\tilde{\boldsymbol{u}}}^0$ as in Equation (3.4), the Schur complement of the global stiffness matrix, $\hat{\boldsymbol{K}}^{gl}$ as in Equation (3.6), as well as $\hat{\boldsymbol{f}}^{gl}$.

(v) Compute the solution for global-local degrees of freedom $\underline{\boldsymbol{u}}^{gl}$ as in Equation (3.7) and total solution u^E as in Equation (3.1).

After the final step, any post-processed quantities which were requested by the user in **GFE-S** are computed, and Abaqus initial global and **GFE-S** local and enriched global solutions may also be visualized. Execution of the program completes, and all requested output from the enriched global solution is available to the user from **GFE-S**.

Chapter 4

Numerical Examples

4.1 L-shaped Domain

The non-intrusive GFEM^{gl} implementation was verified by solving a small, three-dimensional L-shaped domain of overall dimensions $100 \times 100 \times 10$ mm, illustrated in Figure 4.1. The global domain was meshed with tetrahedral elements (TET4 and TET10) 10 mm in size. Nonhomogeneous temperature boundary conditions of $150°$C and $-150°$C were applied to the top and right faces of the domain, respectively, with the rest of the boundary remaining insulated. Thus, this problem is not exposed to any sharp thermal load, as such, but it exhibits a sharp, localized thermal feature in the form of a heat flux singularity at the reentrant inner corner in the domain, making it an ideal verification problem, since no error is incurred due to poor, coarse-scale computation of a sharp load vector.

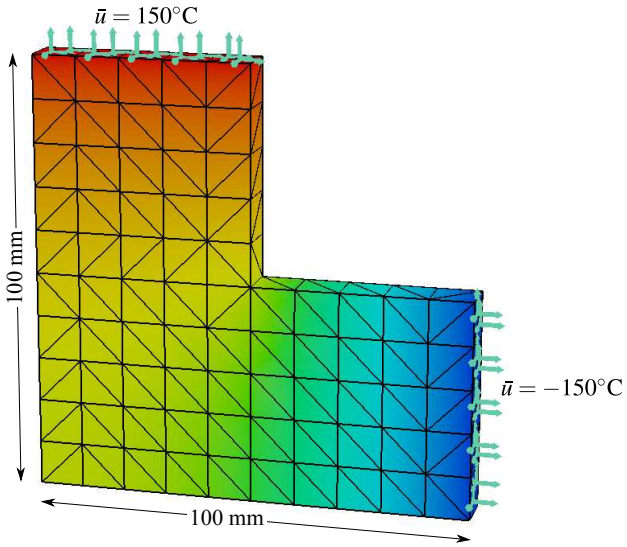


Figure 4.1: L-shaped domain verification problem global model. The enriched global temperature field is illustrated here on the structured tetrahedral mesh used in Abaqus.

Using the GFEM^{gl} methodology, the local problem was chosen as a small neighborhood around the interior corner, $\Omega_L := [20,80] \times [20,80] \times [0,10]$ mm, and the corresponding global-local enrichment zone on the global domain was chosen to be slightly smaller, $X_{gl} := [40,60] \times [40,60] \times [0,10]$ mm. The global and local meshes, as well as the

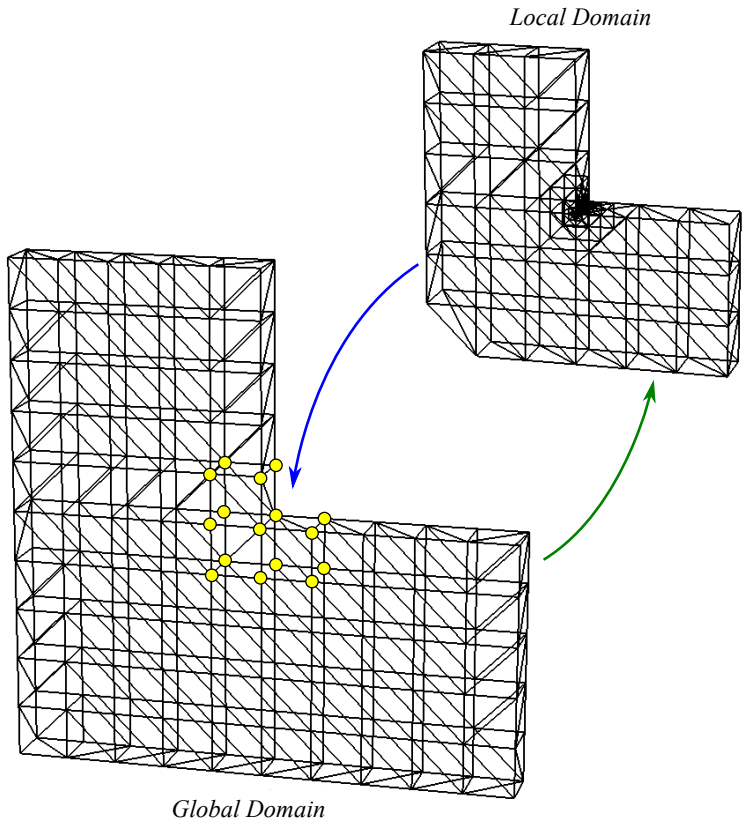


Figure 4.2: Illustration of L-shaped global and extracted local meshes. Geometric mesh refinement is demonstrated in the local domain. Yellow dots on the global domain represent the chosen global-local enrichment zone in the enriched global problem.

global-local enrichment zone, are illustrated in Figure 4.2.

A reference solution u_{ref} was also generated using *hp*-GFEM (the GFEM analog of *hp*-FEM) with 7 levels of global mesh refinement, overall polynomial order $p = 3$, and 25 levels of geometric mesh refinement about the reentrant corner in the domain. Solutions from Abaqus+GFEM^{gl} as well as the standard GFEM^{gl} (without Abaqus) were compared against the *hp*-GFEM reference solution. The two methodologies were found to give nearly identical results; these results are compared in Figure 4.3. Points on the plot were taken from solutions with 4, 10, 16, and 22 levels of geometric refinement on the local mesh. Table 4.1 shows the relative error levels in the solution resulting from simulations using both TET4 and TET10 global meshes, as well as the corresponding number of dofs in each problem. Here, relative error in the energy norm of each finite element solution u_h is computed as

	dofs			e_U^r	
	IG	Local	*EG*	*IG*	*EG*
TET4	192	42,560	192 + 16	11.95%	5.46%
TET10	1,023	42,560	1,023 + 63	4.95%	0.44%

Table 4.1: L-shaped domain computational results for TET4 and TET10 global meshes, 22 levels of *local domain* mesh refinement about the reentrant corner. Energy norm error, e_U^r, is computed with respect to the *hp*-GFEM reference solution. In both cases, the number of additional global-local enrichment dofs is small relative to the global problem size.

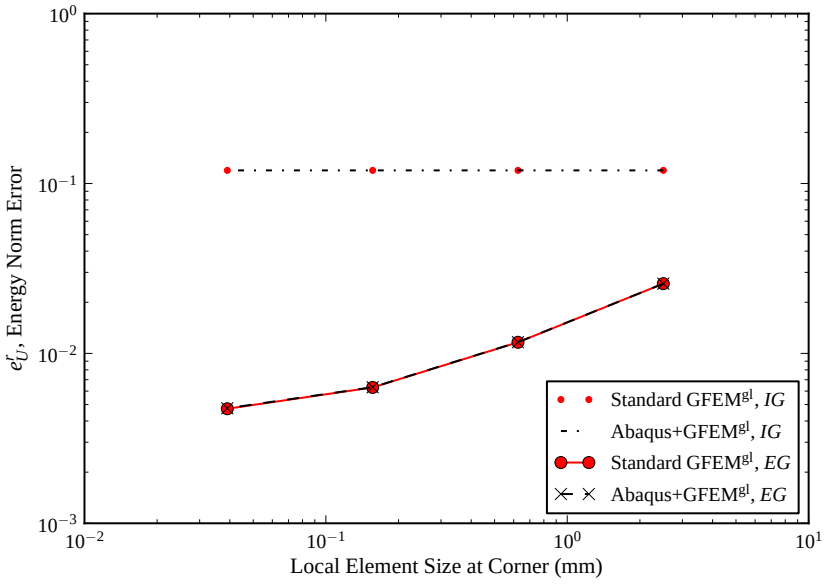
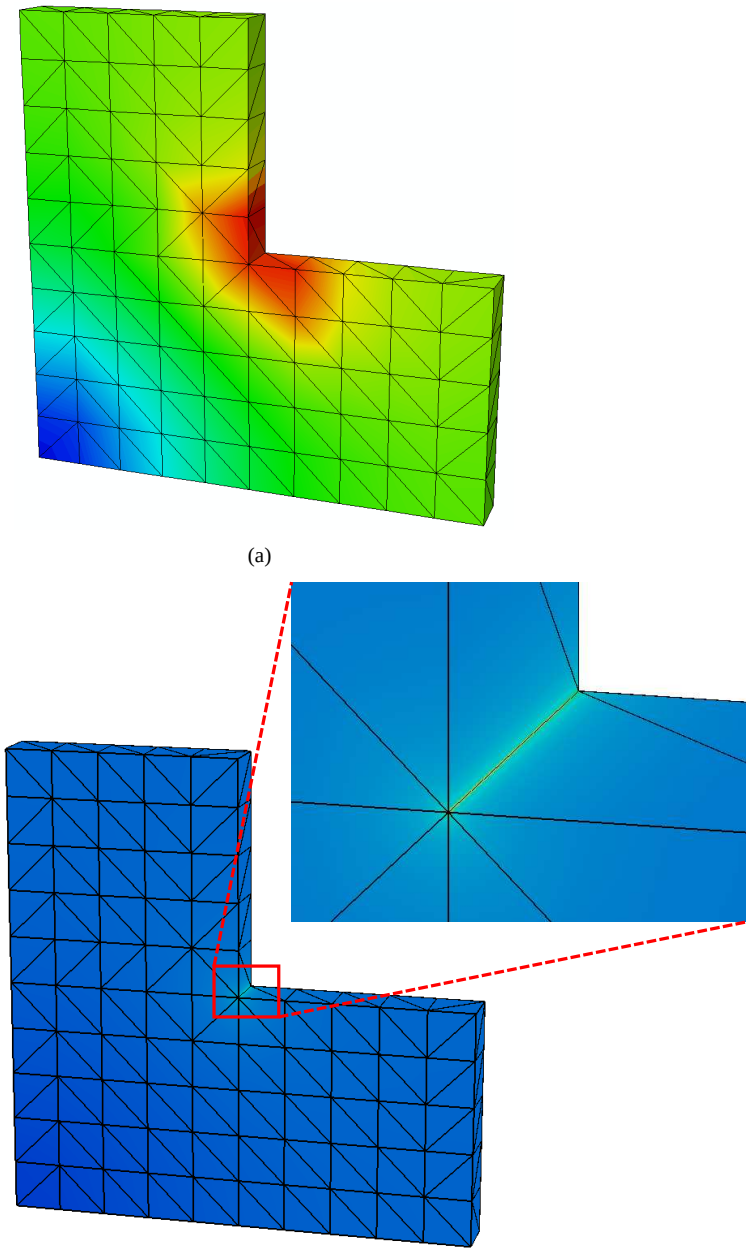


Figure 4.3: L-shaped domain verification problem energy norm errors with respect to the *hp*-GFEM reference solution from both initial global (*IG*) and enriched global (*EG*) solutions, resulting from increasing only the local domain geometric mesh refinement subsequently about the reentrant corner. A quadratic approximation was used in the global problems. Both standard GFEM^{gl} (one solver, without Abaqus) and Abaqus+GFEM^{gl} methodologies were compared for this case, and both were found to give nearly identical results.

$$e_U^r = \sqrt{\frac{\left|B\left(u_{ref},u_{ref}\right) - B\left(u_h,u_h\right)\right|}{\left|B\left(u_{ref},u_{ref}\right)\right|}}. \tag{4.1}$$

While linear TET4 results are poor, the solution can be drastically improved by taking advantage of quadratic TET10 elements in the Abaqus global problem. Based on these results, the corner singularity is resolved very effectively by using Abaqus+GFEM^{gl} with *only* local domain mesh refinement, while adding just a few additional degrees of freedom to the global problem. For qualitative comparison, Figure 4.4 shows the heat flux fields corresponding to initial global and enriched global solutions, respectively, on the TET10 mesh.



(a)

(b)

Figure 4.4: Heat flux field on an L-shaped domain corresponding to (a) Abaqus coarse-scale initial global solution and (b) Abaqus+GFEM^{gl} enriched global solution (1,023 global dofs). The sharp flux resolution possible on a coarse mesh in Abaqus+GFEM^{gl} is also demonstrated here.

4.2 Beam Subjected to Localized Laser Heating

In this example, a sharp, steady-state Gaussian laser flux was applied to the front surface of a small aluminum beam of dimensions $12 \times 0.5 \times 0.24$ inches, illustrated in Figure 4.6. The expression for the flux is given by

$$\bar{f}(\boldsymbol{x}) = I_0 * \frac{1}{2\pi a^2} * G(\boldsymbol{x}, b, a), \qquad 8.0 \leq x \leq 10.0, \tag{4.2}$$

with

$$G(\boldsymbol{x}, b, a) = exp\left(\frac{-(x-b)^2}{2a^2}\right). \tag{4.3}$$

Here, parameter $I_0 = 295 \frac{\text{ft-lbf}}{\text{s}}$ is the laser flux intensity, $a = 0.025$in is the laser focus, or width, and $b = 9.3$in dictates the x-coordinate of the center of the flux. The analytical sharp flux function is shown in Figure 4.5. Convective

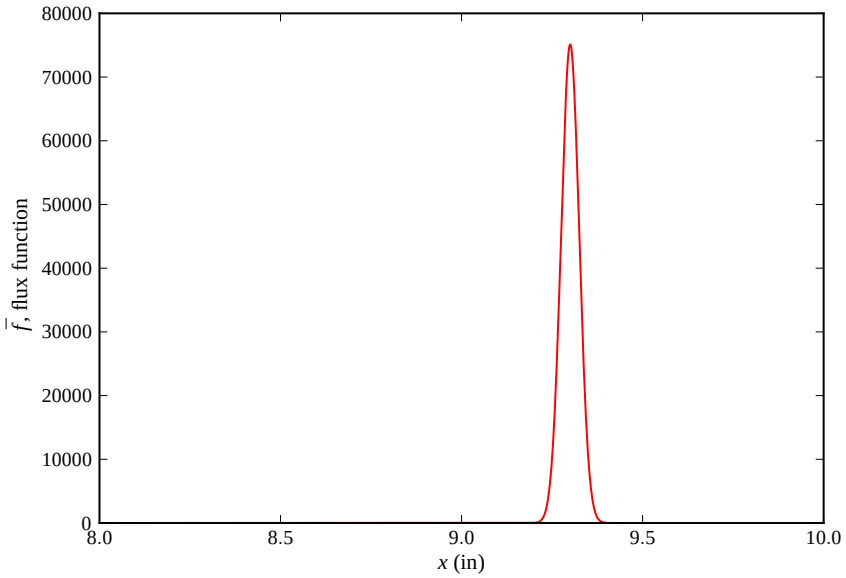


Figure 4.5: Analytical function representing a sharp, Gaussian laser beam heating applied to the front surface, plotted over a small interval on the domain.

conditions were applied on the remainder of the boundary, with convection coefficient $\alpha = 11 \frac{\text{lbf}}{\text{ft-s-}^\circ\text{C}}$ and free-stream temperature $u_\infty = 0^\circ$C.

4.2.1 Convergence Study

Convergence behavior of the proposed Abaqus+GFEM$^{\text{gl}}$ implementation was investigated using this sample problem. A reference solution was developed using *hp*-GFEM with heavy mesh refinement and uniform polynomial order

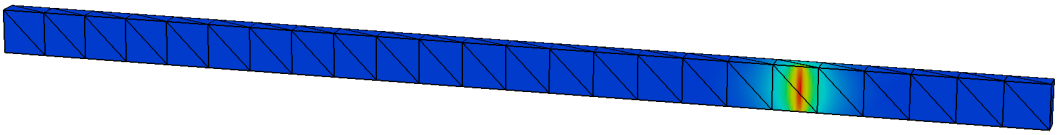


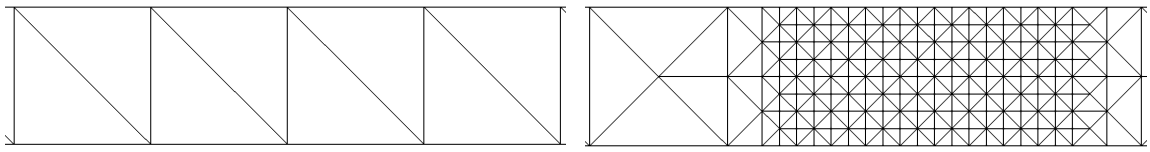
Figure 4.6: Temperature field on a beam subjected to a sharp laser heating. The coarse-scale tetrahedral mesh used in Abaqus is also shown.

$p = 3$ resulting in 736,990 total degrees of freedom. Error was computed as relative error in the energy norm just as in Equation (4.1) using the internal energies of the Abaqus+GFEM^{gl} enriched global and *hp*-GFEM reference solutions, respectively.

The global domain in Abaqus was meshed with TET4 and, subsequently, TET10 elements for comparison. In each case, the global mesh remained constant, consisting of uniform tetrahedral elements 0.5 inches in size (illustrated in Figure 4.6), corresponding to 100 dofs in the TET4 global mesh, and 441 dofs in the TET10 case. The local domain and corresponding enrichment zone also remained a constant size, $\Omega_L := [8.0, 10.0] \times [0.0, 0.5] \times [0.0, 0.24]$ inches, enclosing the entire height and thickness of the beam, where uniform local polynomial order $p = 3$ was used, and a series of local mesh refinements were performed in **GFE-S**. Global-local enrichments in all cases added a mere 20 dofs to the TET4 global problem and 91 dofs to the TET10 global problem; thus, only 20 (TET4) or 91 (TET10) pseudo-loads and pseudo-solutions were exchanged between FEM and GFEM solvers for each mesh refinement level.

For the sake of robustness of the convergence study, the sharp laser heating was intentionally applied such that it was not directly aligned with any global mesh edges, so that the quality of global-local enrichments governed the error and convergence in the enriched global problem solution.

As a comparison, convergence results were also obtained using the optimal methodology, *hp*-GFEM. Each "equivalent" *hp*-GFEM solution resulted from successively refining the mesh near the localized laser heating, analogously to what is done in the GFEM^{gl} local problem, with a uniform polynomial order $p = 3$ throughout. Figure 4.7 illustrates the coarse global mesh used in Abaqus+GFEM^{gl} simulations compared to the very refined global mesh necessitated by equivalent *hp*-GFEM solutions.



(a) (b)

Figure 4.7: Global beam meshes corresponding to (a) Abaqus+GFEM^{gl} and (b) *hp*-GFEM. The enriched global problem in Abaqus+GFEM^{gl} consists of just 120 dofs for the TET4 mesh, or 532 dofs in the TET10 case, whereas the *hp*-GFEM global mesh with $p = 3$ and heavy localized mesh refinement results in a total of 189,290 dofs!

In both Abaqus+GFEM^{gl} and *hp*-GFEM cases, convergence results were computed corresponding to 1, 4, 7, 10,

and 13 levels of mesh refinement in a neighborhood of the sharp laser heating. Figure 4.8 compares convergence of the non-intrusive methodology using Abaqus against the *hp*-GFEM. Relative error in the energy norm, e^r_U, is plotted here

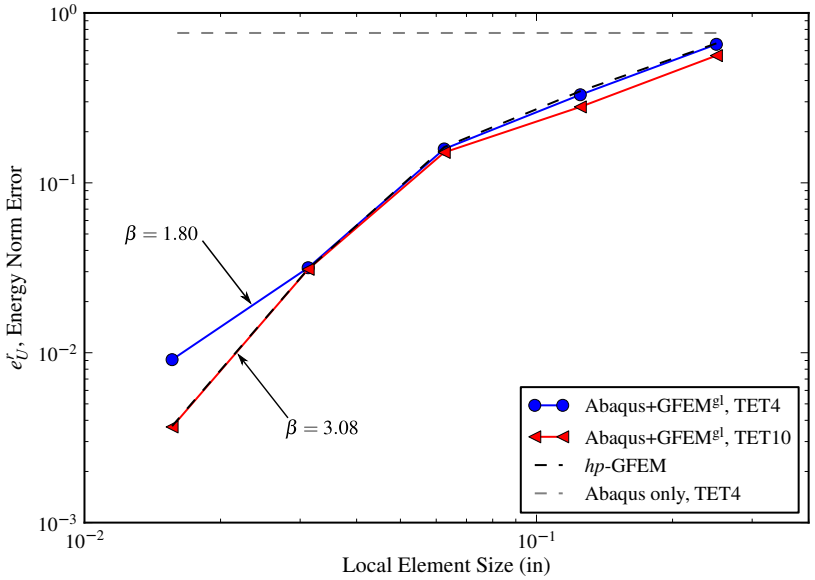


Figure 4.8: Convergence of non-intrusive GFEM^{gl} implementation in Abaqus vs. *hp*-GFEM. Error is computed w.r.t. a reference solution with 736,990 dofs. Asymptotic convergence rates and error levels are nearly identical for Abaqus+GFEM^{gl} (TET10) and *hp*-GFEM. The dashed line shows relative error in the initial global solution, i.e., without any global-local enrichment, on a coarse TET4 mesh using coarse-scale numerical integration of the sharp laser flux (76% error).

against local domain mesh refinement (in the Abaqus+GFEM^{gl} case) or localized global mesh refinement (in the *hp*-GFEM). It should be noted, however, that refinement was *only* performed in the local domain in the Abaqus+GFEM^{gl} case, and that the size of the enriched global problem remained exactly the same for all mesh refinement levels – 100+20 dofs in the TET4 problem, and 441+91 dofs for the TET10 mesh. Results from the Abaqus+GFEM^{gl} methodology are very similar to those obtained using the *hp*-GFEM itself. The convergence rates of the Abaqus+GFEM^{gl} analyses are nearly identical to the *hp*-GFEM, and thus quite near optimal, around the polynomial order of the approximation, $p = 3$. To illustrate the dramatic benefit of using the GFEM^{gl}, the figure also shows the error in the solution obtained from Abaqus using only the coarse global TET4 mesh with no global-local enrichment.

4.2.2 Effect of Special Treatment of the Sharp Loading

Figure 4.9 illustrates the importance of using the GFEM^{gl} local problem information and high-order numerical quadrature rules available only in **GFE-S** to improve accuracy of the global load vector. Convergence results here are given for analyses run in which only coarse-scale computation of the sharp loading on the global mesh was used, and also for cases in which the recomputation procedure (cf. Section 3.2) was used. In both cases, obviously, the sharp loading

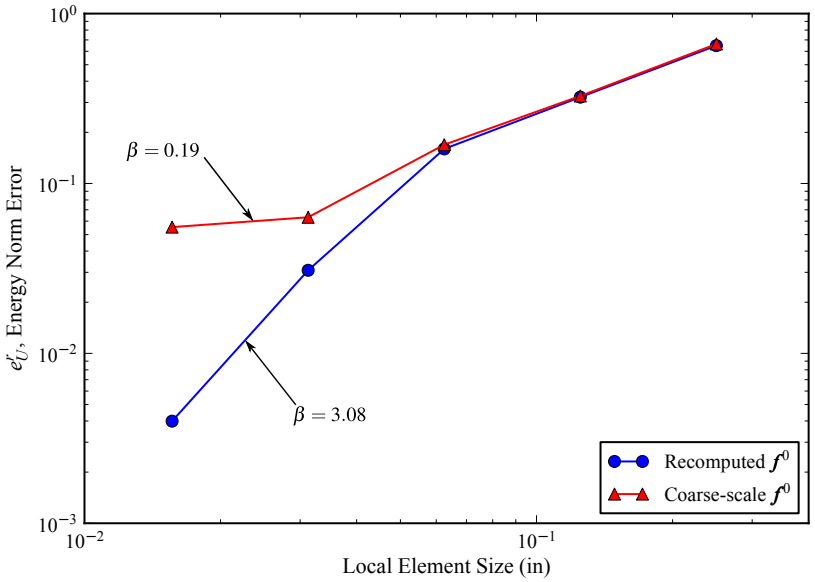


Figure 4.9: Effect of rebuilding the global load vector using GFEM solver with fine-scale computation of the sharp load on convergence behavior of the enriched global solution. Convergence in the energy norm is brought to a halt as the mesh is refined when inaccurate, coarse-scale computation of the sharp loading is performed because representation of the global load is poor, and thus the solution corresponding to global degrees of freedom is similarly unsatisfactory.

was applied in the global domain. Based on the results shown, optimal convergence would, in fact, not nearly be possible without using this scheme for accurately recomputing the global load vector.

Furthermore, an investigation was performed to compare the accuracy of computation of the sharp Gaussian laser flux on a coarse mesh in Abaqus, versus coarse-scale computation of the load using a high-order numerical quadrature rule in the GFEM code, versus the recomputation approach described in Section 3.2, where a high-order quadrature rule is paired with the use of highly-refined local domain elements for integration. The resulting internal energies and error levels are summarized in Table 4.2. Results are shown for both initial and enriched global problems. Here, internal energy is computed as $U_h = 1/2 * B(u_h, u_h)$.

	TET4 mesh		TET10 mesh	
Method used	Int. Energy	e^r_U	Int. Energy	e^r_U
Abaqus, *IG*, coarse-scale	4.036×10^5	92.67%	3.539×10^6	48.84%
GFEM, *IG*, coarse-scale	5.672×10^5	89.53%	1.430×10^6	70.68%
GFEM, *IG*, fine-scale recomputation	5.568×10^5	89.73%	1.439×10^6	70.47%
Abaqus+GFEM^{gl}, *EG*, coarse-scale	2.850×10^6	5.25%	2.860×10^6	2.95%
Abaqus+GFEM^{gl}, *EG*, fine-scale recomputation	2.857×10^6	0.91%	2.857×10^6	0.37%

Table 4.2: Error levels resulting from various approaches to computing the global sharp load vector on the coarse, global mesh. The Abaqus and GFEM coarse-scale computations were performed on the initial global problem, that is, using the initial, uniform, coarse mesh. A higher-order quadrature rule was used in the GFEM computation. Recomputation using highly-refined local elements was carried out in the last *IG* case. Enriched global results corresponding to 13 levels of local domain mesh refinement are also shown for each load computation method.

In the case of the initial global problem in Abaqus, while the TET4 mesh results seem reasonable compared to the **GFE-S** results, the internal energy of the TET10 case is overestimated, and from this it is evident that computation of the sharp loading in Abaqus is unreliable. While Abaqus has implemented a feature enabling the user to define a custom (analytical) loading which may be applied as a concentrated (nodal) or surface load on the model, the methodology is extremely limited by the fact that Abaqus converts the analytical field into some sort of "equivalent," piecewise-constant, averaged surface loading over faces of elements on which it is applied. Thus, large errors are incurred for the problem of interest which exhibits intense and very localized loadings over coarse elements. Evidently, this feature must be used with extreme caution, and it does not work satisfactorily for the methodology used here, justifying the partitioning strategy for the global load vector as described in Section 3.2. In enriched global results, on the other hand, it is shown that an accurate estimate of the actual global load vector is necessary to better approximate the solution corresponding to global dofs and obtain acceptable error levels.

Finally, Figure 4.10 illustrates the effects on the enriched global solution of applying versus *not* applying the coarse-scale, sharp load at all in the initial global problem. Here, the difference in relative error in the energy norm

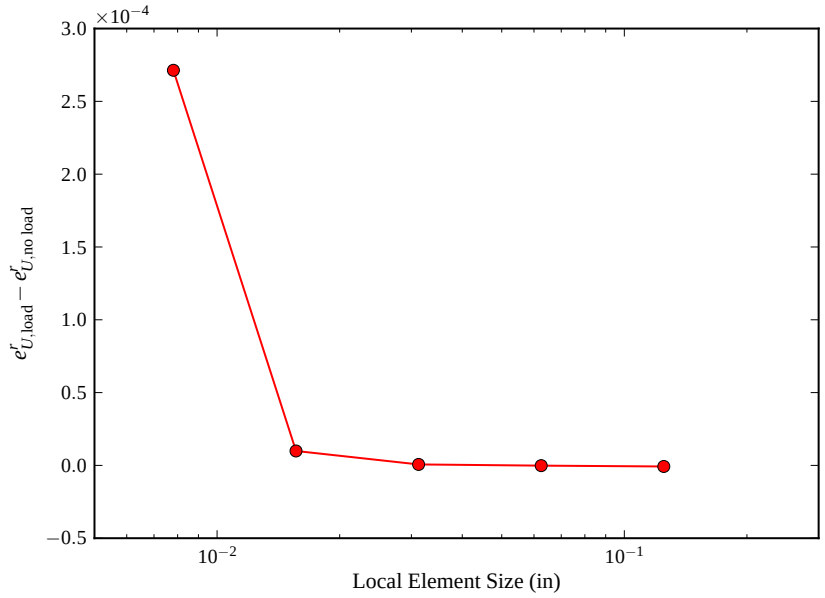


Figure 4.10: Difference in energy norm error of the enriched global beam problem solution between the case where a sharp loading is ($e^r_{U,\text{load}}$) versus is not ($e^r_{U,\text{no load}}$) applied on the coarse, initial global TET10 beam mesh is shown. As the local mesh reaches high levels of refinement, pollution error due to the sharp *IG* problem flux dominates the case where the initial sharp load is applied.

of the enriched global problem solution between the case where the sharp loading is and is not applied is plotted against local domain mesh refinement. Each plot point corresponds to 4, 7, 10, 13, and 16 levels of local domain mesh refinement, respectively, again, with the enriched global problem remaining exactly the same size. It can be observed that as discretization error is reduced as a result of high levels of local mesh refinement, the effect of pollution error

(and some integration error) from applying the sharp flux on the coarse, initial mesh dominates. As shown in Table 4.2, however, the initial solution on a coarse mesh may not improve as a result of decreased integration error.

4.3 Large Stiffened Panel

In the final sample problem presented, an attempt is made to demonstrate the adaptability of the proposed Abaqus+ GFEM^{gl} methodology in handling a variety of different multiscale analysis cases for the same Abaqus global model of interest. This nice feature of the non-intrusive implementation allows the user to insert localized problem information anywhere within the global model, allowing the GFEM^{gl} to handle these localized features, and not requiring any changes to the model itself.

The problem of interest is taken to be a representative, computationally large stiffened panel section of dimensions $600 \times 600 \times 3$ cm with stiffener beams of cross-section 20×20 cm attached underneath the panel, near each edge. The panel was subjected to a constant, steady-state surface heat flux as well as intense, localized Gaussian laser heatings at various locations. The sharp, analytical flux function used to represent the localized heating is plotted in Figure 4.11.

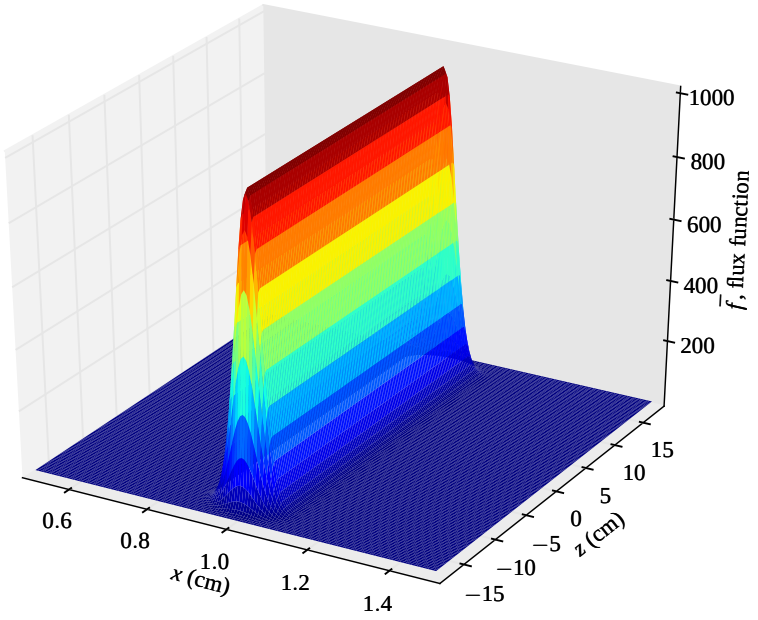


Figure 4.11: Sample depiction of the sharp Gaussian laser flux applied to the top of the panel, plotted over a very small interval on the domain. The laser heating covers an area of approximately 0.1×30 cm on the global domain in all cases.

Adopting the partitioning strategy for $\bar{f}^0$ discussed in Section 3.2, localized laser heatings are introduced *only* in local and enriched global problems, and in all cases, the very same Abaqus global problem was used, upon which only the smooth (constant) portion of the surface heating was applied. Thus, **GFE-S** alone was tasked with handling all computations involving localized effects. In each case, the sharp flux was placed slightly to the left or right, or above

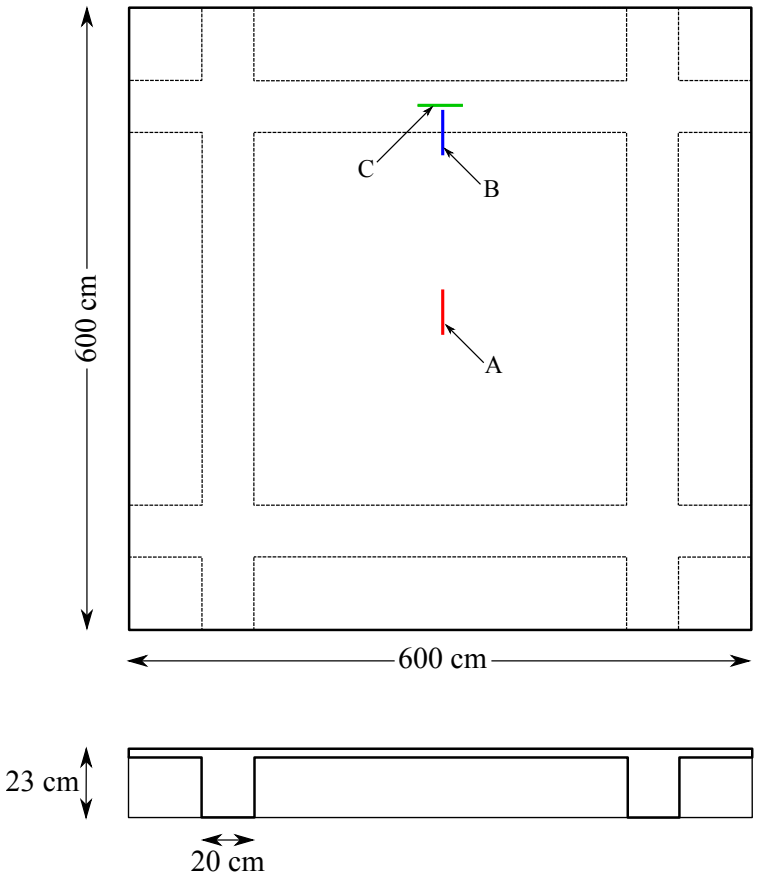


Figure 4.12: Geometry of the panel and locations of applied intense, localized surface laser heating: location A, around the center of the panel; location B, over a panel-stiffener junction; and location C, along the axis of an edge stiffener. *Note: drawing not to scale.*

or below a mesh edge, but never directly in line with one, so that the quality of the GFEM^{gl} solution is dependent strongly upon the quality of the global-local enrichment functions computed in the local problem. This was done to represent the most general possible analysis case, where the mesh may not necessarily be designed to account for the location and configuration of the sharp loading, which may not be known *a-priori*.

The global domain was meshed with linear TET4 elements, resulting in 45,534 initial global degrees of freedom. Temperature boundary conditions of 0°C were applied to the left and right faces of the panel with the boundary insulated elsewhere. The three chosen "critical" locations for the laser heating are illustrated in Figure 4.12, along with the general panel geometry. As usual, in each case, a local domain was chosen to cover a small neighborhood of the locally-applied, sharp laser flux. A sample local domain, in this case, corresponding to sharp flux location A, at the center of the panel, is shown in Figure 4.13. Necessary mesh refinements and polynomial enrichments were taken care of by **GFE-S** only in the local problem, because for this particularly large problem, the cost of refining the global mesh would be much higher than, for example, in the small beam problem used in the convergence study.

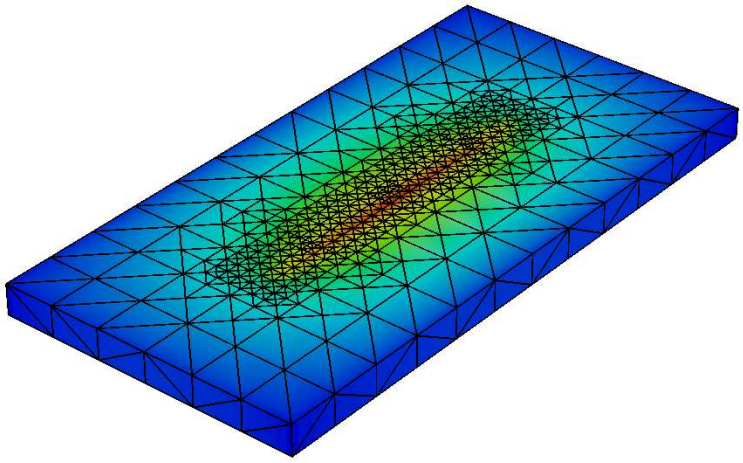


Figure 4.13: Sample local problem shown for sharp flux location A. The temperature distribution on the local domain and refinement about the sharp, localized laser heating are also depicted. The size of the local problem here is $30 \times 60 \times 3$ cm, and in all cases is significantly smaller than the global domain.

	dofs			Internal Energy		
Flux Loc.	*IG*	Local	*EG*	*IG*	Local	*EG*
A	45,534	46,680	45,534 + 55	4.3886×10^6	8.9115×10^4	5.5746×10^6
B	45,534	197,000	45,534 + 85	4.3886×10^6	2.4546×10^7	1.1880×10^8
C	45,534	342,520	45,534 + 85	4.3886×10^6	1.8495×10^7	1.1447×10^8

Table 4.3: Abaqus+GFEM^{gl} numerical results for a stiffened panel problem. The tremendous savings stemming from use of the GFEM^{gl} methodology with respect to additional enriched global problem degrees of freedom is evident here. While *IG* and local solutions underestimate the true solution (drastically in some cases), the enriched global problem is much better able to capture true, sharp solution characteristics.

The initial global temperature field corresponding to a smooth, constant flux over the top surface of the panel is shown in Figure 4.14. Note that because the global model remains unchanged for each sharp load case (only the constant surface flux is applied each time), this same initial global solution may be utilized repeatedly for multiple sharp flux cases as needed, saving some computational cost when many analysis cases are required. Computational data for each sharp flux case is listed in Table 4.3. The corresponding enriched global solutions exhibiting the sharp solution characteristics are shown graphically in Figures 4.15, 4.16, and 4.17.

While the argument may be made that meshes consisting of two-dimensional plate and one-dimensional beam elements may be used in the elastic analysis of structures of a similar nature to this plate in order to reduce computational cost and simplify challenging three-dimensional meshing requirements, in general these types of models may *not* be used in heat transfer analyses. As illustrated by Figure 4.16, the sharp flux of interest applied near the edge of the panel exhibits significant through-the-thickness effects due to the presence of a stiffener beam underneath. Thus, a full, three-dimensional analysis is crucial for predicting actual thermal behavior.

It should be noted that in problems of this nature – large, representative, structural-scale problems which necessitate fine global meshes in order to accurately represent complicated geometry – it would be computationally infeasible

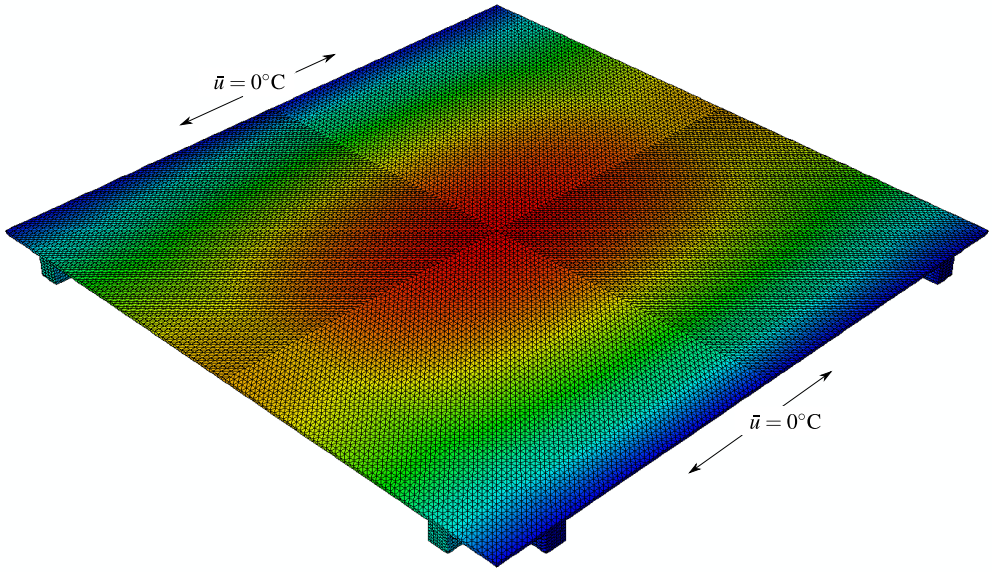


Figure 4.14: Temperature field on a stiffened panel from the Abaqus initial global solution (only a constant surface flux applied). The Abaqus TET4 mesh and temperature boundary conditions are also illustrated here.

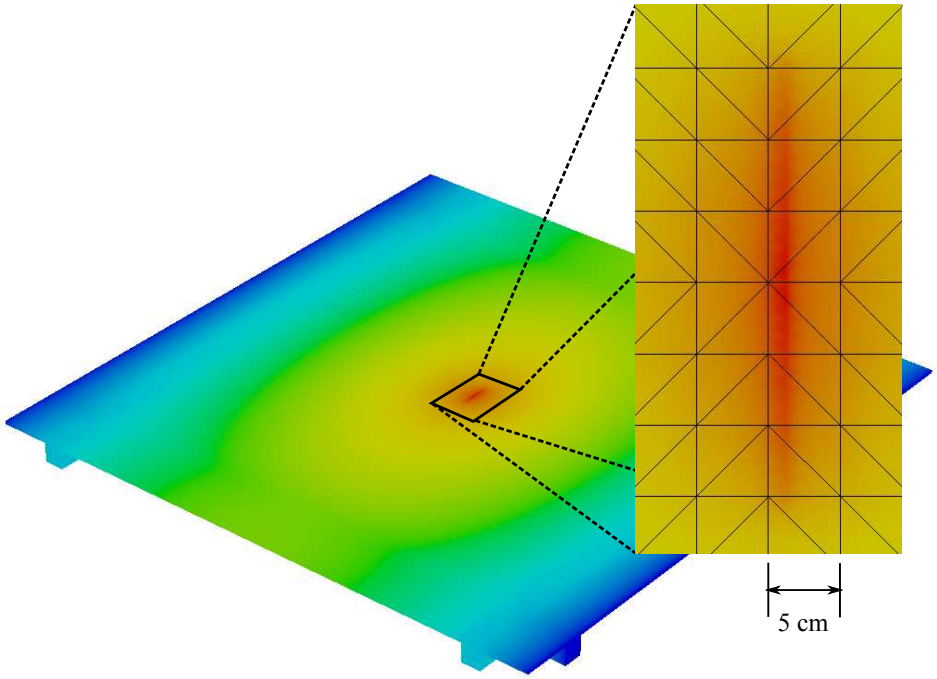


Figure 4.15: Enriched global temperature field on the panel from Abaqus+GFEM^{gl}, flux location A, on the center of the panel. A zoom-in on the sharp feature of interest is also shown. The sharp flux is slightly skewed to the right of the line of a global mesh edge, so that the quality of the sharp global solution feature relies strongly on the quality of global-local enrichments.

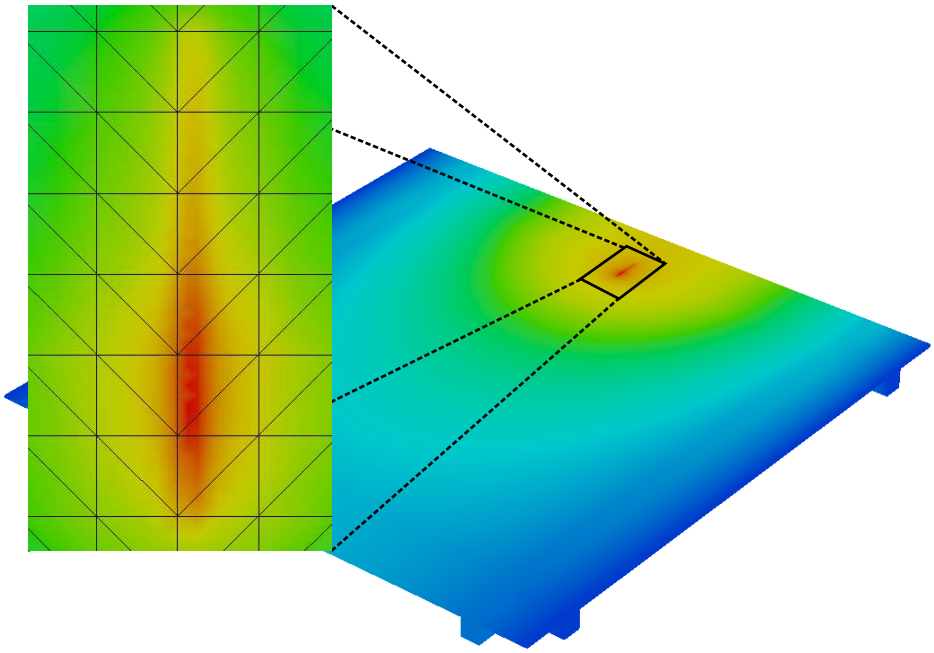


Figure 4.16: Temperature field on a stiffened panel for sharp flux location B. Here, the laser heating intersects a stiffener beam near the edge of the panel. The effectiveness of the 3D structural-scale model in accounting for through-the-thickness effects due to the stiffener is evident here. The sharp flux is slightly skewed to the right of the line of a global mesh edge, so that the quality of the sharp global solution feature relies strongly on the quality of global-local enrichments.

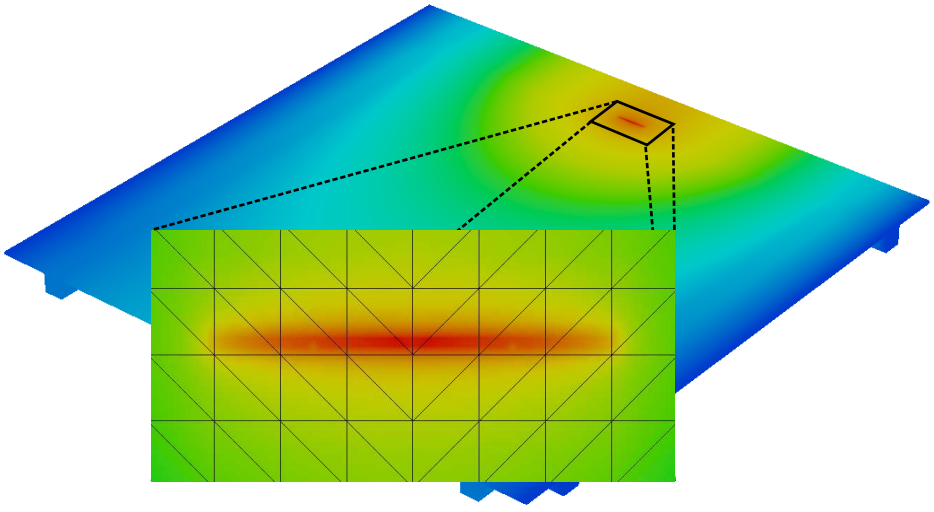


Figure 4.17: Temperature field on a stiffened panel for sharp flux location C, in which case the flux is concentrated entirely over a stiffener beam. The sharp flux is applied slightly above the line of a global mesh edge, so that the quality of the sharp global solution feature relies strongly on the quality of global-local enrichments.

to perform *hp*-adaptivity on the global mesh to capture local solution features. The GFEM^{gl} circumvents this issue entirely, since *hp*-adaptivity need only be performed on a comparatively small subset of the global domain. Without the capability for such *hp*-adaptivity, coarse scale computation of the sharp flux on the global mesh might provide limited prediction of the intense temperature field surrounding the sharp flux on the panel, but the localized, sharp solution characteristics could not be captured satisfactorily.

The immense flexibility of the methodology is also demonstrated clearly by this example problem, since a localized feature can be placed arbitrarily within the global problem without making any changes to the global model itself, which, as aforementioned, is a significant limitation of implementations in Abaqus which utilize, for instance, `UEL`, user element subroutines. Not only can different localized features be inserted into the global problem of interest, but any combination of localized effects can similarly be considered simultaneously via the extraction and solution of multiple local problems from the global domain.

Chapter 5

Conclusions

5.1 Major Benefits of the Abaqus Implementation

The adaptation of state-of-the-art techniques for solving challenging multiscale problems in a legacy FEA code has proven to be a nontrivial task. However, summarizing results from previous sections, the implementation has demonstrated a few largely beneficial characteristics.

5.1.1 Convergence of the Methodology

Employing strategies to provide advanced *hp* finite element discretizations, as well as to allow accurate representation and computation of sharp thermal loadings on structural-scale meshes, the Abaqus+GFEM^{gl} formulation and implementation exhibit a unique and particularly appealing feature-base which has not been identified by the author as being present in any other currently available commercial FEA platform.

In Section 4.2, a sample problem requiring each of the strategies investigated as part of the scope of this study – high-order polynomial enrichments coupled with heavy localized mesh refinement, as well as difficult computation of a sharp loading – was solved. It was demonstrated that the methodology is able to deliver optimal or near-optimal convergence results and error levels on par with the *hp*-GFEM, a strategy that has been recognized as optimal for this class of problems [34, 39]. Most importantly, this is possible without the ability to perform mesh refinements or use high-order polynomial approximations, or to model and compute localized, intense, sharp loads on a coarse mesh in Abaqus.

5.1.2 Flexibility of the Methodology

the Abaqus+GFEM^{gl} implementation provides incomparable flexibility to the user in being able to insert localized effects of interest into a preexisting Abaqus global model at arbitrary locations, which might prove useful in many industrial applications where several analysis cases are required. The possibility for automatic selection of local problems enclosing localized features of interest and adaptive refinement of the mesh, based upon *a-posteriori* error estimates from the initial global solution, u^0, makes the methodology even more appealing for this application.

In Section 4.3, this adaptability was demonstrated on a computationally large, representative problem, in which several different localized, sharp heatings were introduced to the very same global model in Abaqus. In all cases, the enriched global problem solution was able to capture well the expected sharp, localized temperature field, as well as the global-scale temperature field on the domain.

5.2 Limitations of the Abaqus Implementation

Several significant limitations in Abaqus functionalities were encountered while writing and testing the implementation of the Abaqus+GFEM^{gl} framework for heat transfer problems, and the author finds it important to make mention of a few of these limitations in hopes that they might be addressed in future versions of the Abaqus FEA software suite or in future work on non-intrusive GFEM/X-FEM implementations.

5.2.1 Storage of Initial Global Stiffness Matrix

One of the most troublesome limitations of the non-intrusive implementation lies in the fact that the analysis process in Abaqus is not interactive. Specifically, the user can submit a so-called job, wait for completion of said job, and view analysis results after it has completed; however, at no point can the user pause the analysis, store any desired incremental data, and instruct Abaqus to wait or to accept new input for the running job.

As was addressed in the previous section, because the global stiffness matrix, $\boldsymbol{K}^0$, does not change between initial global and enriched global problem steps, it may *theoretically* be stored and saved for the enriched global step. Unfortunately, because of this limitation of Abaqus, this is impossible, and thus the global stiffness matrix must be computed and factorized twice for each Abaqus+GFEM^{gl} analysis, causing some additional computational cost to be incurred when using the non-intrusive implementation. This issue was also explicitly documented in [24].

5.2.2 Multiple Right Hand Sides in Heat Transfer Problems

Another limitation encountered during the implementation of the methodology described here, as mentioned above, is the fact that Abaqus does not have a direct way of handling multiple right hand sides (multiple load cases) for heat transfer problems. While the `*LOAD CASE` command, enabling the specification of multiple right hand sides or load vectors within a single Abaqus analysis step, is implemented for elasticity problems, this option is not available for heat transfer. Thus, a workaround was necessary in order to handle pseudo-loads in each Abaqus+GFEM^{gl} enriched global problem, in which each right hand side is analyzed in a separate steady-state Abaqus load step (`*STEP`); however, this alternative approach complicated the implementation substantially more than in an analogous non-intrusive implementation for elasticity problems.

5.2.3 Enforcement of Boundary Conditions

Because Abaqus employs a "direct" approach to enforcing Dirichlet boundary conditions – different from the implementation of Dirichlet boundary conditions in the GFEM, which uses a penalty method – complications arose in the form of large errors in the enriched global problem when nonhomogeneous boundary conditions were applied on the Abaqus global model. While this also applies to elasticity problems, in general this is a much more important issue in heat transfer, where temperature boundary conditions are most often, in fact, nonzero!

The limitation arises from the fact that in the static condensation algorithm for solving the enriched global problem, the pseudo-load right hand sides have no physical interpretation in the traditional sense. Abaqus, in the presence of nonhomogeneous Dirichlet boundary constraints, in fact *modifies* each pseudo-load right hand side by adding equivalent forces due to the initial nonzero nodal temperatures – that is, given nonhomogeneous temperature boundary conditions $\bar{u}$,

$$u = u_0 + \bar{u},$$

where u_0 is the solution corresponding to homogeneous boundary conditions. The weak form of the problem in Abaqus then becomes

$$B(u,v) = B(u_0 + \bar{u}, v) = L(v),$$

which leads to a modified right hand side:

$$B(u_0,v) = L(v) - B(\bar{u},v)\,.$$

This issue can be avoided altogether by applying the nonhomogeneous boundary conditions as usual in the initial global model, but substituting homogeneous Dirichlet boundary conditions in the enriched global problem, such that $\bar{u} = 0$, and thus the right hand side remains unmodified as desired, yet the constrained degrees of freedom remain the same. This is a strong caveat inherent in using the non-intrusive implementation, since Abaqus does not give the option to solve unmodified right hand sides when nonzero Dirichlet boundary conditions are used.

5.2.4 Abaqus `*CO-SIMULATION` Feature

Abaqus has some very limited built-in facilities to communicate with external, for example, computational fluid dynamics codes, which are accessible to the user through the `*CO-SIMULATION` command. However, as of Version 6.10-1, these facilities are poorly documented and are restricted to use with just a few, proprietary codes. An extension of the `*CO-SIMULATION` framework could be invaluable in allowing for a non-intrusive methodology with fewer workarounds to allow any other external code, such as the GFEM code used in this study, to communicate easily with

Abaqus. Improvement of this command could also resolve the aforementioned issue of having to throw away the factorization of the global stiffness matrix after the initial global step in each Abaqus+GFEM^{gl} simulation.

5.2.5 User-Defined Loads

As documented in Section 4.2, Abaqus provides no acceptable means of applying, for instance, a sharp user-defined loading on a coarse mesh. While the user may define analytical fields and apply them as loadings or boundary conditions on the model, the capabilities of this feature are limited. Specifically, for instance, in the case of a sharp heat flux, Abaqus converts the "analytical field" into an equivalent surface flux by an unknown, internal subroutine. This feature is poorly documented in Abaqus – that is, it is difficult to discern exactly how Abaqus computes the load to be applied – but seems to take an average of the magnitude of the user-defined field over the surface of each affected element. As a result, Abaqus gives a very poor estimate of the true loading, which causes large errors in the global solution.

5.3 Future Work

The promise of non-intrusive implementations of advanced finite element methods in legacy FEA codes like Abaqus is obviously appealing, especially in industrial applications which necessitate challenging multiscale simulations of a similar nature to those introduced here. However, the current implementation is limited to handling only effects of sharp, localized thermal loadings.

As part of work on an upcoming project, the techniques discussed (utilizing the GFEM^{gl}) will be extended to handle advanced multiscale, multi-physics problems, with the next logical step being the introduction of capabilities to handle thermomechanical effects.

References

[1] I. Babuška and J.M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.

[2] I. Babuška and T. Strouboulis. *The Finite Element Method and its Reliability*. Numerical Mathematics and Scientific Computation. Oxford Science Publications, New York, USA, 2001.

[3] I. Babuška, G. Caloz, and J.E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31(4):945–981, 1994.

[4] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulations in Materials Science and Engineering*, 17:24pp, 2009. http://dx.doi.org/10.1088/0965-0393/17/4/043001.

[5] S. Bordas and B. Moran. Enriched finite elements and level sets for damage tolerance assessment of complex structures. *Engineering Fracture Mechanics*, 73:1176–1201, 2006.

[6] Dassault Systèmes Simulia Corporation. Abaqus 6.10, 2010.

[7] D. D'Ambrosio. Numerical prediction of laminar shock/shock interactions in hypersonic flow. *Journal of Spacecraft and Rockets*, 40(2):153–161, 2003.

[8] *Abaqus Analysis User's Manual, Vols. I-V*. Dassault Systèmes Simulia Corporation, Providence, RI, USA, 2010.

[9] *Abaqus/CAE User's Manual*. Dassault Systèmes Simulia Corporation, Providence, RI, USA, 2010.

[10] *Abaqus Scripting User's Manual*. Dassault Systèmes Simulia Corporation, Providence, RI, USA, 2010.

[11] *Abaqus Scripting Reference Manual*. Dassault Systèmes Simulia Corporation, Providence, RI, USA, 2010.

[12] A.Th. Diamantoudis and G.N. Labeas. Stress intensity factors of semi-elliptical surface cracks in pressure vessels by global-local finite element methodology. *Engineering Fracture Mechanics*, 72:1299–1312, 2005.

[13] C.A. Duarte. *The hp Cloud Method*. PhD dissertation, The University of Texas at Austin, December 1996. Austin, TX, USA.

[14] C.A. Duarte and D.-J. Kim. Analysis and applications of a generalized finite element method with global-local enrichment functions. *Computer Methods in Applied Mechanics and Engineering*, 197(6-8):487–504, 2008. doi: 10.1016/j.cma.2007.08.017.

[15] C.A. Duarte, I. Babuška, and J.T. Oden. Generalized finite element methods for three dimensional structural mechanics problems. *Computers and Structures*, 77:215–232, 2000.

[16] C.A. Duarte, O.N. Hamzeh, T.J. Liszka, and W.W. Tworzydlo. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 190(15-17):2227–2262, 2001. doi: 10.1016/S0045-7825(00)00233-4.

[17] C.A.M. Duarte and J.T. Oden. Hp clouds–A meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, May 1995.

[18] C.A.M. Duarte and J.T. Oden. *Hp* clouds – An *hp* meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.

[19] C.A.M. Duarte and J.T. Oden. An *hp* adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139:237–262, 1996.

[20] C.A. Felippa. Introduction to finite element methods., 2004. Course Notes. Department of Aerospace Engineeing Sciences, University of Colorado at Boulder. Available at http://www.colorado.edu/engineering/Aerospace/CAS/courses.d/IFEM.d.

[21] P. P. Friedmann, K. G. Powell, J. J. Mcnamara, B. J. Thuruthimattam, and R. Bartels. Hypersonic aerothermoelastic studies for reusable launch vehicles, 2004. AIAA 2004-1590.

[22] P. P. Friedmann, K. G. Powell, J. J. Mcnamara, B. J. Thuruthimattam, and R. Bartels. Three-dimensional aeroelastic and aerothermoelastic behavior in hypersonic flow, 2005. AIAA 2005-2175.

[23] L. Gendre, O. Allix, P. Gosselet, and F. Comte. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics*, 44(2):233–245, February 2009. doi: 10.1007/s00466-009-0372-9.

[24] L. Gendre, O. Allix, and P. Gosselet. A two-scale approximation of the Schur complement and its use for non-intrusive coupling. *International Journal for Numerical Methods in Engineering*, 2011. doi: 10.1002/nme.

[25] E. Giner, N. Sukumar, J. E. Tarancón, and F. J. Fuenmayor. An Abaqus implementation of the extended finite element method. *Engineering Fracture Mechanics*, 76(3):347–368, February 2009. doi: 10.1016/j.engfracmech.2008.10.015.

[26] V. Gupta, D.-J. Kim, and C.A. Duarte. On the robustness of the generalized finite element method with global-local enrichments. Submitted for publication, 2011.

[27] D.-J. Kim, C.A. Duarte, and J.P. Pereira. Analysis of interacting cracks using the generalized finite element method with global-local enrichment functions. *Journal of Applied Mechanics*, 75(5):051107, 2008. doi: 10.1115/1.2936240.

[28] J.M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[29] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.

[30] J. R. Moselle, A. R. Wieting, M.S. Holden, and C. Glass. Studies of aerothermal loads generated in regions of shock/shock interaction in hypersonic flow, 1988. AIAA-88-0477.

[31] A.K. Noor. Global-local methodologies and their applications to nonlinear analysis. *Finite Elements in Analysis and Design*, 2:333–346, 1986.

[32] J.T. Oden, C.A. Duarte, and O.C. Zienkiewicz. A new cloud-based *hp* finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.

[33] P. O'Hara, C.A. Duarte, and T. Eason. Generalized finite element analysis of three-dimensional heat transfer problems exhibiting sharp thermal gradients. *Computer Methods in Applied Mechanics and Engineering*, 198 (21-26):1857–1871, 2009. doi: 10.1016/j.cma.2008.12.024.

[34] W. Rachowicz, J. T. Oden, and L. Demkowicz. Toward a universal *h-p* adaptive finite element strategy, Part 3. Design of *h-p* meshes. *Computer Methods in Applied Mechanics and Engineering*, 77:181–212, 1989.

[35] J. Shi, J. Lua, H. Waisman, P. Liu, T. Belytschko, N. Sukumar, and Y. Liang. X-FEM toolkit for automated crack onset and growth prediction. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–22, Schaumburg, IL, USA, April 2008.

[36] J. Shi, D. Chopp, J. Lua, N. Sukumar, and T. Belytschko. Abaqus implementation of extended finite element method using a level set representation for three-dimensional fatigue crack growth and life predictions. *Engineering Fracture Mechanics*, 77(14):2840–2863, September 2010. doi: 10.1016/j.engfracmech.2010.06.009.

[37] A. Simone, C.A. Duarte, and E. van der Giessen. A generalized finite element method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering*, 67(8):1122–1145, 2006. doi: 10.1002/nme.1658.

[38] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:4081–4193, 2001.

[39] B. Szabo and I. Babuška. *Finite Element Analysis*. John Wiley and Sons, New York, 1991.

[40] E. A. Thornton. Thermal structures – four decades of progress. *Journal of Aircraft*, 29(3):485–498, 1992. ISSN 0021-8669.

[41] E. A. Thornton, A. R. Wieting, and K. Morgan. Application of integrated fluid-thermal-structural analysis methods. *Journal of Thin-Walled Structures*, 11:1–23, 1991.

[42] T. L. Turner and R. L. Ash. Analysis of the thermal environment and thermal response associated with thermal-acoustic testing, 1990. AIAA-90-0975-CP.

[43] A. R. Wieting. Experimental study of shock wave interference heating on a cylindrical leading edge, 1987. NASA Technical Memorandum 100484.

[44] Y. Xu and H. Yuan. Computational analysis of mixed-mode fatigue crack growth in quasi-brittle materials using extended finite element methods. *Engineering Fracture Mechanics*, 76(2):165–181, January 2009. doi: 10.1016/j.engfracmech.2008.08.011.

[45] Y. Xu and H. Yuan. Computational modeling of mixed-mode fatigue crack growth using extended finite element methods. *International Journal of Fracture*, 159(2):151–165, August 2009. doi: 10.1007/s10704-009-9391-y.